



TUGAS AKHIR - KI141502

IMPLEMENTASI METODE *HYBRID* SALIENCY-SVM UNTUK PEMILIHAN DATA *TRAINING* SECARA OTOMATIS DALAM SEGMENTASI CITRA

**AISHA YULIANDARI
NRP 5111100150**

**Dosen Pembimbing I
Dr.Chastine Fatichah, S.Kom., M.Kom.**

**Dosen Pembimbing II
Rully Soelaiman, S.Kom, M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015**



FINAL PROJECT - KI141502

IMPLEMENTATION HYBRID SALIENCY-SVM METHOD FOR AUTOMATIC TRAINING DATA SELECTION IN IMAGE SEGMENTATION

**AISHA YULIANDARI
NRP 5111100150**

**Supervisor I
Dr.Chastine Fatichah, S.Kom., M.Kom.**

**Supervisor II
Rully Soelaiman, S.Kom, M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2015**

LEMBAR PENGESAHAN

IMPLEMENTASI METODE *HYBRID* SALIENCY-SVM UNTUK PEMILIHAN DATA *TRAINING* SECARA OTOMATIS DALAM SEGMENTASI CITRA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visualisasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
AISHA YULIANDARI
NRP: 5111100150

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Chastine Fatichah, S.Kom., M.Kom.

(pembimbing 1)

Rully Soelaiman, S.Kom, M.Kom.

(pembimbing 2)

SURABAYA
JUNI, 2015

IMPLEMENTASI METODE *HYBRID* SALIENCY-SVM UNTUK PEMILIHAN DATA *TRAINING* SECARA OTOMATIS DALAM SEGMENTASI CITRA

Nama Mahasiswa : AISHA YULIANDARI
NRP : 5111100150
Jurusan : Teknik Informatika FTIF-ITS
**Dosen Pembimbing 1 : Dr.Chastine Fatichah, S.Kom.,
M.Kom.**
Dosen Pembimbing 2 : Rully Soelaiman, S.Kom., M.Kom.

Abstrak

Segmentasi citra adalah salah satu tahapan penting dalam visi komputer dan pengolahan citra, yang nantinya akan digunakan untuk temu kembali citra, pengenalan objek dan klasifikasi data. Segmentasi citra bisa dilihat sebagai masalah klasifikasi, yaitu dengan menandai masing-masing piksel menurut ciri-ciri tertentu.

Support Vector Machine (SVM) adalah metode klasifikasi yang termasuk dalam supervised learning. Supervised learning merupakan metode yang membutuhkan training dan testing. Training sample yang digunakan pada proses training tidak selalu ada pada beberapa kasus, terutama kasus segmentasi citra.

Penelitian ini mengimplementasikan metode berbasis SVM yaitu Saliency-SVM untuk pemilihan data training secara otomatis dalam segmentasi citra. Metode ini menghasilkan data training menggunakan visual saliency berbasis SVM di mana terdapat tahapan pra-segmentasi dan pembentukan trimap berdasarkan informasi saliency dari visual saliency detection, kuantisasi ruang warna HSV, analisis histogram dan local homogeneity threshold. Data training yang dihasilkan adalah piksel yang termasuk positif(objek) dan negatif(background). Tahap sebelum dilakukan

segmentasi dengan SVM adalah ekstraksi fitur untuk menghasilkan vektor input pada SVM. Segmentasi objek pada citra dilakukan dengan SVM berdasarkan SVM Trained Model.

Hasil uji coba dari Saliency-SVM untuk segmentasi citra ini memiliki nilai rata-rata akurasi mencapai 94,84% dibandingkan dengan citra ground truth.

Kata kunci: Segmentasi citra, SVM, visual saliency detection, pemilihan data training.

IMPLEMENTATION HYBRID SALIENCY-SVM METHOD FOR AUTOMATIC TRAINING DATA SELECTION IN IMAGE SEGMENTATION

Student's Name : AISHA YULIANDARI
Student's ID : 5111100150
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr.Chastine Fatichah, S.Kom.,
M.Kom.
Second Advisor : Rully Soelaiman, S.Kom., M.Kom.

Abstract

Image segmentation is one important step in computer vision and image processing, which will be used form image retrieval, object recognition, and data classification. Image segmentation can be seen as classification problem, namely by marking each pixel with certain characteristics.

Support Vector Machine (SVM) is a classification method that is included in supervised learning. Supervised learning is a method that requires training and testing. Training samples which used in learning process does not always exist in some cases, especially in image segmentation cases.

This research implements SVM bases method that is Saliency-SVM for choose training data automatically in image segmentation. This method generates training data using visual saliency based SVM, where there are pre-segmentation and trimap generation based on saliency information of visual saliency detection, HSV color space quantization , histogram analysis and local homogeneity threshold. Training data is generated pixels which including positive (object) and negative (background).Step before use segmentation with SVM is feature extraction to generate

input vectors in SVM. Object segmentation in image is done by SVM based SVM Trained Model.

The test result of saliency-SVM for image segmentation has an average value accuracy reached 94.84% by comparison premises ground truth image.

Keywords: Image segmentation, SVM, Visual saliency detection, data training selection.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul **“IMPLEMENTASI MOETODE HYBRID SALIENCY-SVM UNTUK PEMILIHAN DATA TRAINING SECARA OTOMATIS DALAM SEGMENTASI CITRA”**.

Harapan dari penulis semoga apa yang tertulis di dalam penelitian ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata bagi kampus Teknik Informatika, ITS, dan bangsa Indonesia.

Dalam pelaksanaan dan pembuatan penelitian ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT dan Nabi Muhammad SAW, karena atas limpahan rahmat-Nya, penulis diberikan kemudahan dan kelancaran dalam mengerjakan penelitian ini.
2. Kedua orang tua penulis, Bapak Erwin Romel dan Ibu Gustiwarni atas segala dukungan moril, spiritual dan material, dengan memberi semangat, perhatian, dan selalu sabar dalam menghadapi penulis yang selalu mengeluh, serta memberikan doa yang tiada habisnya untuk penulis.
3. Bapak Rully Soelaiman, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan motivasi, nasehat, bimbingan, bantuan serta menjadi teman berbagi cerita selama penulis mengerjakan penelitian ini.
4. Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom selaku dosen pembimbing yang telah memberikan motivasi, nasehat, bimbingan, bantuan serta mendengarkan keluhan dari penulis selama mengerjakan penelitian ini.

5. Ibu Dr. Nanik Suciati selaku ketua jurusan Teknik Informatika ITS, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Bapak dan Ibu staf Tata Usaha yang telah memberikan bantuan dan kemudahan kepada penulis selama masa perkuliahan di Teknik Informatika ITS.
7. Kakak-kakak dari penulis yaitu Dicky Rizki Romel dan Randa Gusti Romel yang telah memberikan semangat, dan berbagi pengalaman kepada penulis.
8. Teman-teman yang sudah sabar menghadapi pertanyaan-pertanyaan penulis pada saat mengerjakan Tugas Akhir ini.
9. Teman-teman sepermainan penulis yang terpisah antara pulau-pulau Rossa Amelia, Fitriyani, Ayu syahnaztria, Annisa Kurnia, Annisa Aulia, Khairunnisa, Mayang Kania, Edlyn Shella, Revina, Mentari dan Nurul yang telah mendukung dan menghibur penulis.
10. Teman-teman angkatan 2011 yang menemani, membantu dan memotivasi selama penulis sampai pada tahap pengerjaan penelitian ini.
11. Bapak Purnomo yang sudah sabar mengantar-jemput penulis selama ini.
12. Juga tak lupa kepada semua pihak yang belum sempat disebutkan satu per satu di sini yang telah membantu dalam penyelesaian penelitian ini.

Kesempurnaan tentu masih belum tercapai pada tugas akhir ini. Karena itu, penulis mengharapkan saran dan kritik dari pembaca untuk perbaikan selanjutnya.

Surabaya, Juni 2015

Aisha Yuliandari

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i> vii	
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxiii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi	3
1.7 Sistematika Penulisan.....	4
2 BAB II TINJAUAN PUSTAKA.....	7
2.1 Segmentasi Citra.....	7
2.2 Tahapan Umum Segmentasi Citra Berbasis Informasi <i>Saliency</i>	7
2.3 Pemilihan Data <i>Training</i> dengan <i>Visual Saliency Detection</i> berbasis SVM	8
2.3.1 <i>Visual Saliency Detection</i> dengan Pendekatan <i>Spectral</i> <i>Residual</i> 9	
2.3.2 Binerisasi Citra dengan Metode Otsu	12
2.3.3 Operasi Morfologi	13
2.3.4 Kuantisasi Ruang Warna HSV	15
2.3.5 Analisis Histogram	19
2.3.6 <i>Local Homogeneity Threshold</i>	23
2.4 Ekstraksi Fitur	24
2.4.1 Fitur Warna	24
2.4.2 Fitur <i>Saliency</i>	25
2.4.3 Fitur Spasial	25

2.4.4	Fitur Tekstur	25
2.5	Klasifikasi Piksel dengan <i>Support Vector Machine</i>	29
2.5.1	Normalisasi Data.....	29
2.5.2	Klasifikasi dengan SVM	29
2.5.3	Metode Evaluasi.....	32
3	BAB III ANALISIS DAN PERANCANGAN.....	33
3.1	Perancangan Secara Umum	34
3.2	Perancangan Pemilihan <i>Training</i> Piksel dengan <i>Visual Saliency</i> berbasis SVM.....	35
3.2.1	Perancangan Pra-segmentasi dan Pembentukan <i>Trimap</i>	36
3.2.2	Kuantisasi Ruang Warna HSV.....	36
3.2.3	Perancangan Analisis Histogram	37
3.2.4	Perancangan Pemilihan <i>Training</i> Piksel dengan <i>Local Homogeneity Threshold</i>	38
3.3	Perancangan Ekstraksi Fitur	39
3.3.1	Perancangan Fitur Warna.....	39
3.3.2	Perancangan Fitur <i>Saliency</i>	40
3.3.3	Perancangan Fitur Spasial.....	40
3.3.4	Perancangan Fitur Tekstur	40
3.4	Perancangan Klasifikasi Piksel dengan SVM.....	41
3.4.1	Perancangan Normalisasi Data	42
3.4.2	Perancangan <i>Training</i> SVM	42
3.4.3	Perancangan <i>Testing</i> SVM.....	42
3.4.4	Perancangan Evaluasi	42
3.5	Perancangan Antarmuka Aplikasi	44
4	BAB IV IMPLEMENTASI.....	47
4.1	Lingkungan Implementasi	47
4.2	Implementasi	47
4.2.1	Implementasi Secara Umum	47
4.2.2	Implementasi Pemilihan <i>Training</i> Piksel dengan <i>Visual Saliency</i> Berbasis SVM.....	49
4.2.3	Implementasi Ekstraksi Fitur	54
4.2.4	Implementasi Klasifikasi Piksel dengan SVM.....	58
4.3	Implementasi Antarmuka	60

5	BAB V PENGUJIAN DAN EVALUASI	63
5.1	Lingkungan Uji Coba	63
5.2	Data Uji Coba	63
5.3	Skenario dan Evaluasi Pengujian	64
5.3.1	Uji coba dan Evaluasi Skenario 1	64
5.3.2	Uji Coba dan Evaluasi Skenario 2	67
5.3.3	Uji Coba dan Evaluasi Skenario 3	69
5.3.4	Uji Coba dan Evaluasi Skenario 4	71
5.3.5	Uji Coba dan Evaluasi Skenario 5	72
	BAB VI KESIMPULAN DAN SARAN	75
6.1.	Kesimpulan	75
6.2.	Saran	76
	DAFTAR PUSTAKA	77
	LAMPIRAN	79
	BIODATA PENULIS	129

DAFTAR TABEL

Tabel 2.1 Fungsi Kernel yang Biasa Digunakan dalam SVM.....	32
Tabel 3.1 Variabel yang Digunakan pada Perancangan (Bagian 1)	33
Tabel 3.2 Variabel yang Digunakan pada Perancangan (Bagian 2)	34
Tabel 5.1 Parameter dan Metode Awal (Bagian 1)	64
Tabel 5.2 Parameter dan Metode Awal (Bagian 2)	65
Tabel 5.3 Hasil Uji Coba Skenario 1 (Bagian 1)	65
Tabel 5.4 Hasil Uji Coba Skenario 1 (Bagian 2)	66
Tabel 5.5 Hasil Uji Coba Skenario 2 (Bagian 1)	67
Tabel 5.6 Hasil Uji Coba Skenario 2 (Bagian 2)	68
Tabel 5.7 Hasil Uji Coba Skenario 2 (Bagian 3)	68
Tabel 5.8 Hasil Rata-Rata Akurasi Uji Coba Skenario 2	69
Tabel 5.9 Hasil Uji Coba Skenario 3	70
Tabel 5.10 Hasil Uji Coba Skenario 4	71
Tabel 5.11 Hasil Rata-Rata Akurasi Uji Coba Skenario 3	72
Tabel 5.12 Hasil Segmentasi Uji Coba Skenario 5 (Bagian 1) ...	72
Tabel 5.13 Hasil Segmentasi Uji Coba Skenario 5 (Bagian 2) ...	72
Tabel 5.14 Hasil Segmentasi Uji Coba Skenario 5 (Bagian 3) ...	72
Tabel 5.15 Hasil Rata-Rata Akurasi Uji Coba Skenario 5	73
 Tabel A. 1 Citra Masukan (Bagian 1)	 79
Tabel A. 2 Citra Masukan (Bagian 2)	80
Tabel A. 3 Hasil Segmentasi Uji Coba 1 (Bagian 1)	80
Tabel A. 4 Hasil Segmentasi Uji Coba 1 (Bagian 2)	81
Tabel A. 5 Hasil Citra dari Tahapan-Tahapan (Bagian 1)	82
Tabel A. 6 Hasil Citra dari Tahapan-Tahapan (Bagian 2)	83
Tabel A. 7 Hasil Citra dari Tahapan-Tahapan (Bagian 3)	84
Tabel A. 8 Hasil Citra dari Tahapan-Tahapan (Bagian 4)	85
Tabel A. 9 Hasil Citra dari Tahapan-Tahapan (Bagian 5)	86
Tabel A. 10 Hasil Citra dari Tahapan-Tahapan (Bagian 6)	87
Tabel A. 11 Hasil Citra dari Tahapan-Tahapan (Bagian 7)	88
Tabel A. 12 Hasil Citra dari Tahapan-Tahapan (bagian 8)	89

Tabel A. 13 Hasil Citra dari Tahapan-Tahapan (Bagian 9).....	90
Tabel A. 14 Hasil Citra dari Tahapan-Tahapan (Bagian 10).....	91
Tabel A. 15 Hasil Citra dari Tahapan-Tahapan (Bagian 11).....	92
Tabel A. 16 Hasil Citra dari Tahapan-Tahapan (Bagian 13).....	93
Tabel A. 17 Hasil Citra dari Tahapan-Tahapan (Bagian 14).....	94
Tabel A. 18 Hasil Citra dari Tahapan-Tahapan (Bagian 15).....	95
Tabel A. 19 Hasil Citra dari Tahapan-Tahapan (Bagian 16).....	96
Tabel A. 20 Hasil Citra dari Tahapan-Tahapan (Bagian 17).....	97
Tabel A. 21 Hasil Citra dari Tahapan-Tahapan (Bagian 18).....	98
Tabel A. 22 Hasil Citra dari Tahapan-Tahapan (Bagian 19).....	99
Tabel A. 23 Hasil Citra dari Tahapan-Tahapan (Bagian 20).....	100
Tabel A. 24 Hasil Citra dari Tahapan-Tahapan (Bagian 21).....	101
Tabel A. 25 Hasil Citra dari Tahapan-Tahapan (Bagian 21).....	102
Tabel A. 26 Hasil Citra dari Tahapan-Tahapan (Bagian 22).....	103
Tabel A. 27 Hasil Citra dari Tahapan-Tahapan (Bagian 23).....	104
Tabel A. 28 Hasil Citra dari Tahapan-Tahapan (Bagian 24).....	105
Tabel A. 29 Hasil Citra dari Tahapan-Tahapan (Bagian 25).....	106
Tabel A. 30 Akurasi Uji Coba Skenario 1	107
Tabel A. 31 Hasil Akurasi Uji Coba Skenario 2 (Bagian 1) – <i>bilinear</i> $n=5$	108
Tabel A. 32 Hasil Akurasi Uji Coba Skenario 2 (Bagian 2) – <i>bilinear</i> $n=7$	109
Tabel A. 33 Hasil Akurasi Uji Coba Skenario 2 (Bagian 3) – <i>bicubic</i> $n = 3$	110
Tabel A. 34 Hasil Akurasi Uji Coba Skenario 2 (Bagian 4) – <i>bicubic</i> $n=5$	111
Tabel A. 35 Hasil Akurasi Uji Coba Skenario 2 (Bagian 5) – <i>bicubic</i> $n= 7$	112
Tabel A. 36 Hasil Akurasi Uji Coba Skenario 3 (Bagian 1) – lebar piksel = 70	113
Tabel A. 37 Hasil Akurasi Uji Coba Skenario 3 (Bagian 2) –lebar piksel = 75	114
Tabel A. 38 Data <i>Training</i> Sesuai Nilai <i>Threshold</i>	115
Tabel A. 39 Hasil Akurasi Uji Coba Skenario 4 (Bagian 1) – <i>Threshold</i> = 0	116

Tabel A. 40 Hasil Akurasi Uji Coba Skenario 4 (Bagian 2) – <i>Threshold</i> = 50	117
Tabel A. 41 Hasil Akurasi Uji Coba Skenario 4 (Bagian 3) – <i>Threshold</i> = 100	118
Tabel A. 42 Hasil Akurasi Uji Coba 5 (Bagian 1) –linear C 0.1	119
Tabel A. 43 Hasil Akurasi Uji Coba 5 (Bagian 2) –polynomial(3) C 0.1	120
Tabel A. 44 Hasil Akurasi Uji Coba Skenario 5 (Bagian 3) – polynomial(4) C 0.1	121
Tabel A. 45 Hasil Akurasi Uji Coba Skenario 5 (Bagian 4) – rbf(0.25) C 0.1	122
Tabel A. 46 Hasil Akurasi Uji Coba Skenario 5 (Bagian 5) – rbf(0.5) C 0.1	123
Tabel A. 47 Hasil Akurasi Uji Coba Skenario 5 (Bagian 6) –linear C 1	124
Tabel A. 48 Hasil Akurasi Uji Coba Skenario 5 (Bagian 7) – polynomial(3) C 1	125
Tabel A. 49 Hasil Akurasi Uji Coba Skenario 5 (Bagian 8) – polynomial(4) C 1	126
Tabel A. 50 Hasil Akurasi Uji Coba Skenario 5 (Bagian 9) – rbf(0.25) C 1	127
Tabel A. 51 Hasil Akurasi Uji Coba Skenario 5 (Bagian 10) – rbf(0.5) C 1	128

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Tahapan Umum Segmentasi Citra Berbasis Informasi <i>Saliency</i>	8
Gambar 2.2 Ilustrasi Pra-Segmentasi dan Pembentukan <i>Trimap</i> ..	9
Gambar 2.3. Log Spectrum dari Sebuah Citra	11
Gambar 2.4 <i>Spectral Residual</i> dari Sebuah Citra	11
Gambar 2.5 Perbedaan Hasil <i>Saliency Map</i> dengan Skala Besar dan Skala Kecil	12
Gambar 2.6 Hasil Binerisasi <i>Saliency Map</i> Sebuah Citra	13
Gambar 2.7 Hasil Erosi Terhadap Ro (Mo)	14
Gambar 2.8 Hasil Operasi Dilasi Terhadap Ro (Mb).....	15
Gambar 2.9 Ilustrasi Kuantisasi Ruang Warna HSV	15
Gambar 2.10 Algoritma Konversi Ruang Warna RGB ke HSV ..	16
Gambar 2.11 Skema Kuantisasi <i>Hue Channel</i> [6].....	17
Gambar 2.12 Skema Kuantisasi <i>S&V Channel</i> [6]	18
Gambar 2.13 Ilustrasi Analisis Histogram (Bagian <i>To</i>)	21
Gambar 2.14 Ilustrasi Analisis Histogram (Bagian <i>Tb</i>)	21
Gambar 2.15 (a) <i>Global histogram</i> (b) <i>Pho</i> (c) <i>Pno</i>	22
Gambar 2.16 (a) Posisi <i>TSp</i> (b) Posisi <i>TSn</i>	23
Gambar 2.17 <i>TSp</i> Setelah Dilakukan <i>Local Homogeneity Threshold</i>	24
Gambar 2.18 <i>TSn</i> Setelah Dilakukan <i>Local Homogeneity Threshold</i>	24
Gambar 2.19 Matriks Turunan Gaussian dengan $\sigma=5$ [8]	28
Gambar 2.20 Fitur Tekstur (a) Orientasi 0 (b) Orientasi 30 (c) Orientasi 60 (d) Orientasi 90 (e) Orientasi 120 (f) Orientasi 150 (g) <i>local energy</i> (h) <i>local gradient</i>	28
Gambar 3.1 Diagram Alir Program Secara Umum	35
Gambar 3.2 Diagram Alir Pemilihan <i>Training</i> Piksel dengan <i>Visual Saliency</i> Berbasis SVM.....	36
Gambar 3.3 <i>Pseudocode</i> Pra-segmentasi dan Pembentukan <i>trimap</i>	37
Gambar 3.4 <i>Pseudocode</i> Kuantisasi Ruang Warna HSV	37
Gambar 3.5 <i>Pseudocode</i> Analisis Histogram.....	38

Gambar 3.6 <i>Pseudocode Local Homogeneity Threshold</i>	38
Gambar 3.7 : Diagram alir ekstraksi fitur secara umum.....	39
Gambar 3.8 <i>Pseudocode</i> Ekstraksi Fitur Warna.....	40
Gambar 3.9 <i>Pseudocode</i> Ekstraksi Fitur <i>Saliency</i>	41
Gambar 3.10 <i>Pseudocode</i> Ekstraksi Fitur Spasial.....	41
Gambar 3.11 <i>Pseudocode</i> Ekstraksi Fitur Tekstur	41
Gambar 3.12 Diagram Alir Klasifikasi Piksel dengan SVM Secara Umum.....	43
Gambar 3.13 <i>Pseudocode Training</i> SVM	43
Gambar 3.14 <i>Pseudocode Testing</i> SVM.....	43
Gambar 3.15 Tampilan Rancangan Antarmuka Aplikasi.....	45
Gambar 4.1 Hasil Implementasi Antarmuka	61

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode Implementasi Saliency-SVM Secara Umum (Bagian 1).....	48
Kode Sumber 4.2 Kode Implementasi Saliency-SVM Secara Umum (Bagian 2).....	49
Kode Sumber 4.3 Kode Implementasi Pra-segmentasi dan Pembentukan <i>Trimap</i>	50
Kode Sumber 4.4 Kode Implementasi Fungsi <i>Spectral Residual</i>	51
Kode Sumber 4.5 Kode Implementasi Fungsi Pembentukan <i>Trimap</i>	51
Kode Sumber 4.6 Kode Implementasi Ruang Warna HSV.....	52
Kode Sumber 4.7 Kode Implementasi Analisis Histogram (Bagian 1)	53
Kode Sumber 4.8 Kode Implementasi Analisis Histogram (Bagian 2)	53
Kode Sumber 4.9 Kode Implementasi Analisis Histogram (Bagian 3)	54
Kode Sumber 4.10 Kode Implementasi Ekstraksi Fitur Warna ..	55
Kode Sumber 4.11 Kode Implementasi Ekstraksi Fitur Warna (<i>Intensity</i>)	55
Kode Sumber 4.12 Kode Implementasi Ekstraksi Fitur <i>Saliency</i>	56
Kode Sumber 4.13 Kode Implementasi Ekstraksi Fitur Spasial pada <i>Training</i>	56
Kode Sumber 4.14 Kode Implementasi Ekstraksi Fitur Spasial pada <i>Testing</i>	56
Kode Sumber 4.15 Kode Implementasi Membentuk Nilai <i>Local Energy</i> dan <i>Local Gradient</i>	57
Kode Sumber 4.16 Kode Implementasi <i>Steerable Filter</i> [9].....	57
Kode Sumber 4.17 Kode Implementasi Normalisasi Data	58
Kode Sumber 4.18 Kode Implementasi Pemanggilan Fungsi <i>svmtrain</i>	58
Kode Sumber 4.19 Kode Implementasi Pemanggilan Fungsi <i>svmclassify</i>	59

Kode Sumber 4.20 Kode Implementasi Membentuk Citra Biner
(Hasil Segmentasi).....59
Kode Sumber 4.21 Kode Implementasi Perhitungan Akurasi60

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari penelitian. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan penelitian.

1.1 Latar Belakang

Segmentasi citra adalah salah satu tahapan penting dalam visi komputer dan pengolahan citra, yang nantinya akan digunakan untuk temu kembali citra, pengenalan objek, dan klasifikasi data. Segmentasi citra bisa dilihat sebagai masalah klasifikasi, yaitu dengan menandai masing-masing piksel menurut ciri-ciri tertentu. Beberapa metode-metode klasifikasi sudah berhasil melakukan segmentasi citra, diantaranya metode berbasis *Support Vector Machine* (SVM).

SVM merupakan metode klasifikasi yang termasuk dalam *supervised learning*. *Supervised learning* adalah metode yang membutuhkan *training* dan *testing* [1]. SVM menggunakan beberapa *training sample* yang sudah ditandai untuk pembelajaran klasifikasi, sedangkan *training sample* yang sudah ditandai tidak selalu ada pada beberapa kasus, terutama kasus segmentasi citra.

Maka dari itu dalam penelitian ini akan mengimplementasikan pendekatan baru yang memadukan antara *Visual Saliency Detection* dan SVM *classifier* untuk segmentasi citra. Metode ini menghasilkan data *training* menggunakan *visual saliency* berbasis SVM, di mana terdapat tahapan pra-segmentasi dan pembentukan *trimap* berdasarkan informasi *saliency* dari *visual saliency detection*, kuantisasi ruang warna HSV, analisis histogram dan *local homogeneity threshold*. Setelah data *training* terbentuk dilakukan ekstraksi fitur dan klasifikasi dengan SVM.

Hasil yang diharapkan pada implementasi Saliency-SVM ini adalah dapat melakukan segmentasi citra dengan baik dan menghindari proses pemilihan *training sample* secara manual.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini dapat dipaparkan sebagai berikut :

1. Menggunakan metode *visual saliency detection* dalam pemilihan data *training*.
2. Mengimplementasikan metode *visual saliency detection* dan *SVM classifier* untuk segmentasi citra.
3. Mengevaluasi kinerja pendekatan baru Saliency-SVM untuk segmentasi citra.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam penelitian ini memiliki beberapa batasan antara lain :

1. *Dataset* yang digunakan adalah citra berwarna yang diambil dari [2].
2. *Visual saliency detection* menggunakan pendekatan *spectral residual* [3].
3. Fungsi kernel yang digunakan pada klasifikasi piksel dengan SVM adalah linear, polinomial, dan rbf.

1.4 Tujuan

Tujuan dari pembuatan penelitian ini adalah melakukan implementasi metode *visual saliency detection* dan *SVM classifier* untuk pemilihan data *training* secara otomatis dalam segmentasi citra.

1.5 Manfaat

Manfaat dari penelitian ini adalah mengeksplorasi kinerja klasifikasi pada *SVM classifier* dan menghindari proses pemilihan data *training* dilakukan secara manual.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan penelitian ini adalah sebagai berikut:

1. Penyusunan proposal penelitian.

Tahap awal untuk memulai pengerjaan penelitian ini adalah penyusunan proposal penelitian. Proposal penelitian yang diajukan memiliki gagasan yang sama dengan penelitian ini, yaitu implementasi metode *hybrid* Saliency-SVM untuk pemilihan data *training* secara otomatis dalam segmentasi citra.

2. Studi literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran, dan pemahaman informasi dan literatur yang berhubungan dengan algoritma yang digunakan dalam pengerjaan penelitian ini. Diantaranya mengenai *visual saliency detection*, ruang warna HSV, analisis histogram, *local homogeneity threshold*, dan *SVM classifier*.

3. Analisis dan desain perangkat lunak

Pada tahap ini berisi analisis dan desain perangkat lunak terhadap segmentasi citra. Di mana menganalisis permasalahan yang diangkat pada penelitian ini dan merancang perangkat lunak dengan menentukan data yang digunakan serta proses-proses yang dilakukan dalam penelitian ini.

4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya dengan menggunakan Matlab. Implementasi perangkat lunak menggunakan GUI(*Graphical User Interface*). Di mana pengguna dapat

memasukkan citra dan mendapatkan hasil segmentasi dari citra tersebut.

5. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba pada data citra. Pengujian dan evaluasi akan dilakukan dengan membandingkan hasil segmentasi dengan *ground truth*.

6. Penyusunan Buku

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung penelitian ini.

Bab III Analisis dan Perancangan

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan penelitian.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Bab ini berisi proses implementasi dari setiap kelas pada semua modul.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian *error rate* untuk mengetahui akurasi dari metode yang digunakan jika dibandingkan dengan *ground truth*.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan metode Saliency-SVM yang diimplementasikan pada program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

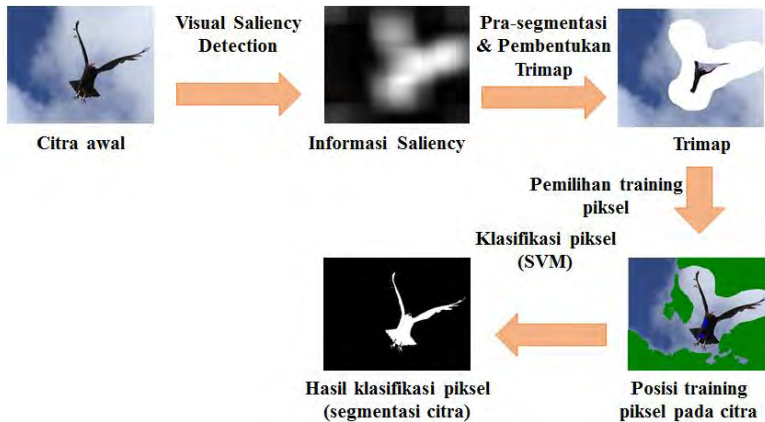
2.1 Segmentasi Citra

Segmentasi citra adalah proses yang membagi citra digital menjadi beberapa segmen (set piksel). Tujuan dari segmentasi citra adalah untuk menyederhanakan atau mengubah representasi citra menjadi sesuatu yang lebih bermakna dan mudah dianalisis. Segmentasi citra biasanya digunakan untuk menemukan objek dan batas-batas dalam citra. Segmentasi citra adalah proses untuk memberi label pada setiap piksel dalam citra sehingga piksel dengan label yang sama saling berbagi karakteristik visual tertentu.

Hasil dari segmentasi citra adalah seperangkat segmen yang secara kolektif mencakup keseluruhan dari gambar. Setiap piksel dalam suatu daerah yang sama memiliki hubungan dengan beberapa properti karakteristik, seperti warna, intensitas atau tekstur [4].

2.2 Tahapan Umum Segmentasi Citra Berbasis Informasi *Saliency*

Informasi *saliency* adalah *saliency map* yang dihasilkan oleh *visual saliency detection*. *Saliency map* berperan untuk menghasilkan perkiraan objek(*foreground*) dari sebuah citra. Tahapan umum dari segmentasi citra berbasis informasi *saliency* ditunjukkan pada Gambar 2.1. Di mana dari informasi *saliency* tersebut, dihasilkan *trimap* yang nantinya akan menghasilkan data *training* yaitu lokasi-lokasi *training* piksel positif(objek) dan *training* piksel negatif(*background*). Data *training* yang dihasilkan tersebut digunakan untuk klasifikasi piksel pada SVM yang nantinya akan menghasilkan segmentasi citra.



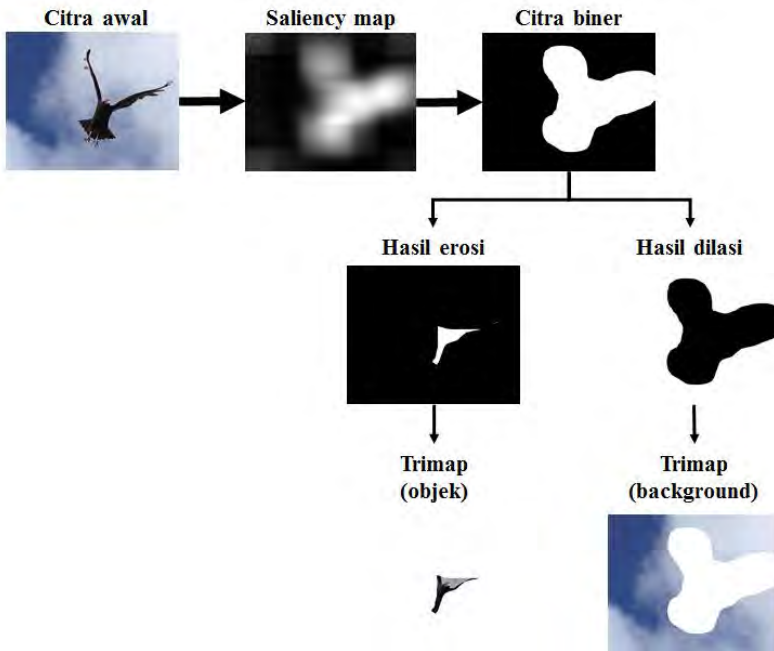
Gambar 2.1 Ilustrasi Tahapan Umum Segmentasi Citra Berbasis Informasi Saliency

2.3 Pemilihan Data *Training* dengan *Visual Saliency Detection* berbasis SVM

Pemilihan *training* piksel dengan *Visual Saliency* berbasis SVM ini meliputi proses pra-segmentasi dan pembentukan *trimap* ditunjukkan oleh Gambar 2.2. Di mana untuk memperkirakan lokasi objek dan *background* berdasarkan *saliency map* yang dihasilkan oleh *visual saliency detection* pada citra masukan. Hasil proses pra-segmentasi dan pembentukan *trimap* adalah citra *trimap* (T_o , T_r dan T_b). T_o adalah bagian yang menonjol pada proses pra-segmentasi, T_b adalah *background* dan T_r adalah piksel-piksel samar dari daerah sisa T_o dan T_b . Perbedaan dasar antara T_o dan T_b adalah fitur warna dan fitur spasial, untuk memilih set piksel dari T_o dan T_b maka digunakan piksel warna dan fitur spasial sebagai pembeda ciri. Jadi piksel-piksel yang memenuhi kriteria sebagai set piksel merupakan SVM *training dataset* (TS_p dan TS_n). Kriterianya adalah sebagai berikut :

- Kriteria spasial yaitu lokasi piksel pada T_o atau T_b .
- Kriteria warna yaitu piksel dengan warna dominan pada T_o atau T_b .

T_o dan T_b yang dihasilkan akan diidentifikasi lebih lanjut dengan analisis histogram untuk pemilihan warna dominan yang sebelumnya sudah dikuantisasi dengan ruang warna HSV, dan *local homogeneity threshold*. Identifikasi tersebut dilakukan untuk menentukan apakah piksel-piksel tersebut termasuk TS_p (*positive training set*) dan TS_n (*negative training set*).



Gambar 2.2 Ilustrasi Pra-Segmentasi dan Pembentukan *Trimap*

2.3.1 *Visual Saliency Detection* dengan Pendekatan *Spectral Residual*

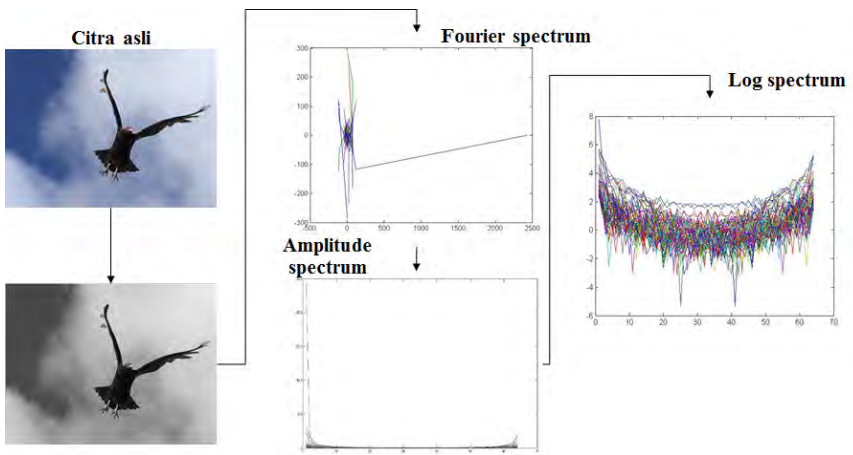
Visual saliency detection dengan pendekatan *spectral residual* adalah sebuah mekanisme untuk membentuk *saliency map*. *Saliency map* yang dihasilkan digunakan untuk menemukan perkiraan objek

dari sebuah citra. *Saliency map* dapat dihasilkan dengan mengaplikasikan Persamaan (2. 1) terhadap *Fourier spectrum*.

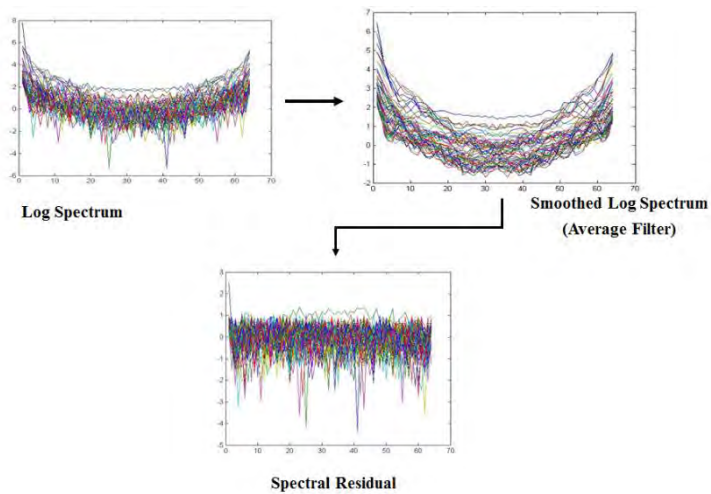
$$\begin{aligned}
 A(f) &= \text{absolute}(F(I(x))) \\
 P(f) &= \text{angle}(F(I(x))) \\
 L(f) &= \log(A(f)) \\
 R(f) &= L(f) - h_n(f) * L(f) \\
 S(f) &= g(x) * ||F^{-1}[\exp R(f) + i * P(f)]||^2
 \end{aligned} \tag{2. 1}$$

$L(f)$ adalah *log spectrum* dari sebuah citra yang ditunjukkan pada Gambar 2.3. Di mana *Log spectrum* dari sebuah citra dihasilkan oleh $A(f)$ yaitu *amplitude spectrum*. Dari *log spectrum* yang dihasilkan maka *spectral residual* $R(f)$ dapat dibentuk dengan menghitung sisa dari *log spectrum* dan *average filter* $h_n(f)$ dari *log spectrum* tersebut, dapat dilihat pada Gambar 2.4. Dari *spectral residual* yang dihasilkan *saliency map* $S(f)$ dapat dibentuk sesuai dengan Persamaan (2. 1) Di mana F^{-1} adalah *Inverse Fourier transform*, $P(f)$ adalah *phase spectrum* dan $g(x)$ adalah *2D Gaussian Filter* dengan $\sigma = 8$ untuk efek visual yang lebih baik. Nilai n pada *average filter* yaitu besar matriks konvolusi adalah 3.

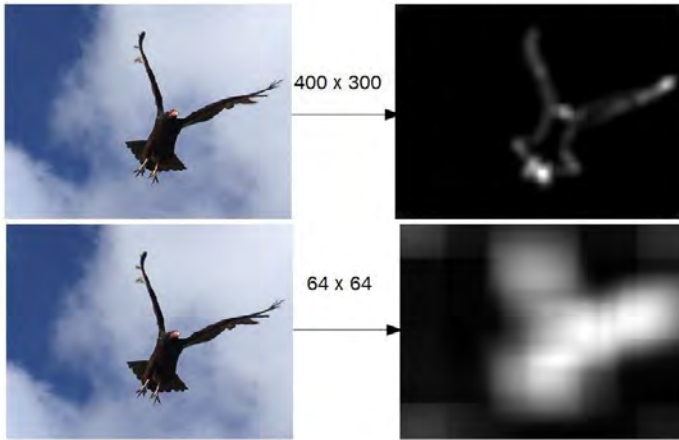
Saliency map dibentuk dari citra dengan skala yang kecil, maka dari itu sebelum *saliency map* dibentuk, sebuah citra di *resize* ke skala 64x64. Pemilihan skala pada *visual saliency detection* mempengaruhi *saliency map* yang dihasilkan. Untuk citra dengan skala kecil dapat menghasilkan *saliency map* untuk objek besar dalam citra tersebut, sedangkan pada skala besar hanya menghasilkan *saliency map* untuk objek kecil dalam citra [3]. Perbedaan hasil *saliency map* dengan skala kecil dan dengan skala besar ditunjukkan oleh Gambar 2.5. Hasil *saliency map* yang dihasilkan oleh *visual saliency detection* sangat mempengaruhi hasil *training set* yang nantinya akan dihasilkan oleh analisis histogram untuk menemukan warna dominan. Maka dari itu dilakukan percobaan-percobaan agar mendapatkan *saliency map* yang sesuai untuk pemilihan *training set*.



Gambar 2.3. Log Spectrum dari Sebuah Citra



Gambar 2.4 *Spectral Residual* dari Sebuah Citra



Gambar 2.5 Perbedaan Hasil *Saliency Map* dengan Skala Besar dan Skala Kecil

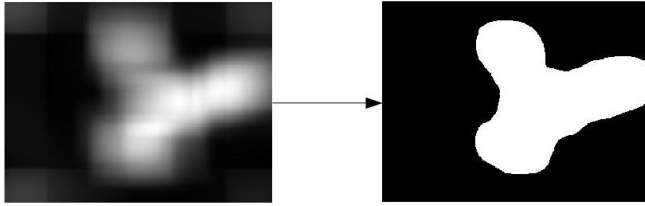
2.3.2 Binerisasi Citra dengan Metode Otsu

Untuk mengidentifikasi lokasi dari objek, dilakukan binerisasi citra dengan metode otsu dengan Persamaan (2. 2)

$$OM(x,y) = \begin{cases} 0 & \text{if } SM(x,y) < t \\ 1 & \text{if } SM(x,y) \geq t \end{cases} \quad (2. 2)$$

Threshold dari binerisasi ini ditentukan oleh kriteria diskriminasi yaitu (σ_B^2, σ_W^2) dari 2 kelas yaitu (*salient object* dan *background*). Binerisasi citra dengan metode otsu sudah diimplementasikan pada matlab dengan toolbox [5].

Di mana memilih *threshold* untuk meminimalkan perbedaan antara piksel hitam dan putih. Untuk bagian putih (objek) yang bernilai 1 adalah R_o sedangkan piksel hitam yang bernilai 0 adalah R_b yang ditunjukkan pada Gambar 2.6.



Gambar 2.6 Hasil Binerisasi *Saliency Map* Sebuah Citra

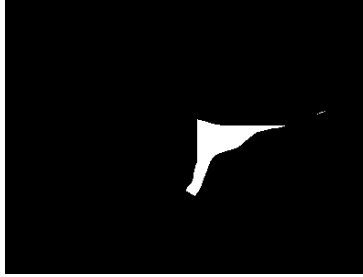
2.3.3 Operasi Morfologi

Morfologi citra biner adalah pemrosesan citra yang merubah bentuk atau struktur objek dalam citra. Ada 3 dasar operasi yaitu dilasi, erosi dan penulangan. Dilasi adalah memproses objek menjadi membesar dalam ekstensi spasial. Erosi adalah memproses objek menjadi menyusut secara *uniform*. Sedangkan penulangan adalah memproses objek menjadi terkikis sampai menjadi kerangka.

Dalam Saliency-SVM ini menggunakan operasi morfologi terhadap citra biner yang telah dihasilkan dari binerisasi citra. R_o adalah objek yang bernilai 1 dari hasil binerisasi citra, dan R_b adalah background yaitu wilayah yang bernilai 0 pada citra biner. Untuk menghilangkan piksel yang tidak penting di sekitar *boundary* R_o maka dilakukan operasi morfologi dengan Persamaan (2. 3)

$$M_o = R_o \ominus E_{r_e} \quad (2. 3)$$

Boundary R_o disusutkan dengan operasi erosi. M_o adalah Mask Object dan $\ominus E_{r_e}$ adalah operator erosi yang menunjukkan penyusutan R_o pada piksel r_e . Hasil operasi erosi terhadap R_o (M_o) dapat ditunjukkan pada Gambar 2.7.



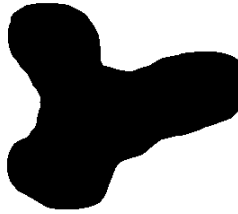
Gambar 2.7 Hasil Erosi Terhadap R_o (M_o)

Persamaan (2. 4) menunjukkan bahwa R_o diperluas dengan operasi dilasi. M_b adalah Mask Background dan $\oplus D_{r_d}$ operator dilasi yang menunjukkan perluasan R_o pada piksel r_d . Hasil dilasi dari R_o dikurang dengan R_o dan hasilnya digabungkan dengan R_b .

$$M_b = ((R_o \oplus D_{r_d}) - R_o) \cup R_b \quad (2. 4)$$

Hasil dari proses dilasi terhadap R_o (M_b) dapat ditunjukkan pada Gambar 2.8. M_o dan M_b yang dihasilkan dari operasi morfologi menjadi patokan lokasi pada pembentukan *trimap* yang sudah dijelaskan pada bab 2.3 dan pada ilustrasi Gambar 2.2. Di mana $Trimap(T_o, T_r$ dan $T_b)$ dalam ruang warna RGB tidak lagi dalam biner.

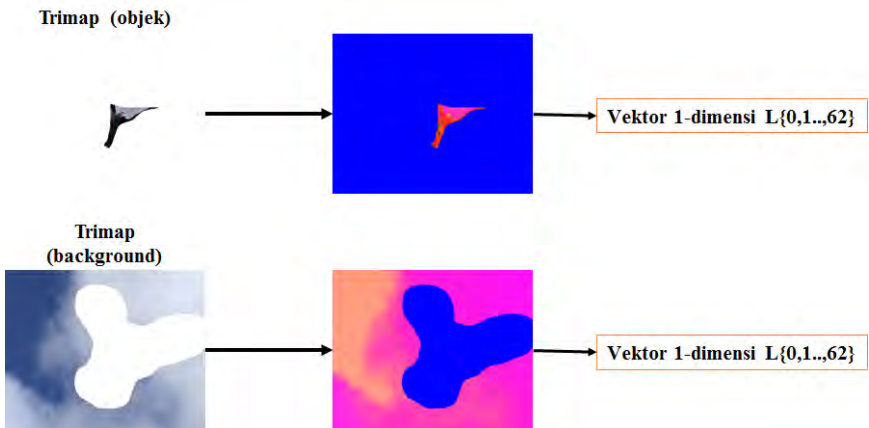
Elemen struktur yang digunakan pada operasi morfologi ini adalah *square*, dengan lebar 70 piksel. Penentuan lebar 70 piksel adalah dengan percobaan yang dilakukan, di mana jika lebar dibawah 70 piksel pada saat analisis histogram untuk memilih warna dominan, warna dominan yang terdapat pada T_o dan T_b bernilai sama. Di mana jika memiliki warna dominan yang sama maka warna dominan tidak dapat menjadi ciri pembeda antara T_o dan T_b seperti yang sudah dijelaskan pada bab 2.3.



Gambar 2.8 Hasil Operasi Dilasi Terhadap Ro (Mb)

2.3.4 Kuantisasi Ruang Warna HSV

Setelah pembentukan $Trimap(T_o, T_r \text{ dan } T_b)$, bagian $Trimap T_o$ dan T_b saja yang diolah pada proses selanjutnya. Karena T_r adalah bagian sisa dari T_o dan T_b maka tidak penting untuk proses selanjutnya. T_o dan T_b dalam ruang warna RGB dikonversi menjadi ruang warna HSV dan dikuantisasi untuk menghasilkan vector 1-dimensi seperti ilustrasi pada Gambar 2.9.



Gambar 2.9 Ilustrasi Kuantisasi Ruang Warna HSV

HSV (*Hue*: $[0,360]$, *Saturation*: $[0,1]$, dan *Value*: $[0,1]$) mampu menekankan pada persepsi visual manusia, yang terbukti memiliki hasil yang lebih baik untuk segmentasi citra dari ruang warna RGB. Algoritma untuk konversi RGB menjadi HSV ditunjukkan oleh Gambar 2.10.

Untuk menurunkan kompleksitas komputasi dalam penentuan warna dominan T_o dan T_b pada analisis histogram maka dilakukan kuantisasi ruang warna HSV pada T_o dan T_b .

Karena sistem visual manusia lebih sensitif untuk *hue* (corak) daripada *saturation* dan *value* maka *hue channel* dikuantisasi lebih halus daripada *saturation* dan *value*. Diketahui bahwa distribusi warna (merah, jingga, kuning, hijau, *cyan*, biru, dan ungu) pada *hue channel* tidak *uniform*, maka akan diaplikasikan skema *non-uniform quantization*.

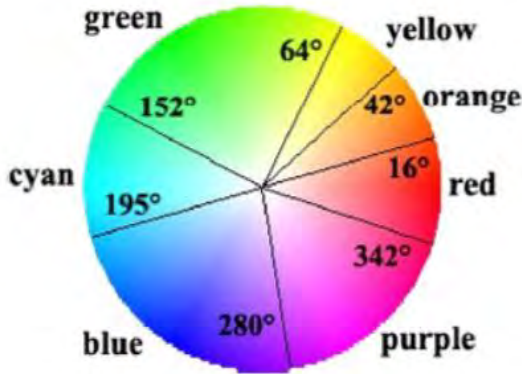
Input	R,G, dan B
Output	H,S dan V
1.	$V := \max(R, G, B);$
2.	$Let X := \min(R, G, B);$
3.	$S := \frac{V-X}{V}, if S = 0 return;$
4.	$Let r := \frac{V-R}{V-X}; g := \frac{V-G}{V-X}; b := \frac{V-B}{V-X};$
5.	$If R = V then H := (if G = X then 5 + b else 1 - g);$
6.	$if G = V then H := (if B = X then 1 + r else 3 - b);$
7.	$else H := (if R = X then 3 + g else 5 - r);$
8.	$H := \frac{H}{6}$

Gambar 2.10 Algoritma Konversi Ruang Warna RGB ke HSV

Gambar 2.11 menunjukkan hasil skema yang sudah dikuantisasi menjadi 7 *non-uniform bins* yang diwakili dari 0-6 dan masing-masing menunjukkan warna utama dengan Persamaan (2. 5).

Gambar 2.12 menunjukkan *saturation* (S) dan *value* (V) *channel* dikuantisasi secara *non-uniform* untuk 3 *bins* dengan cara yang sama, misalnya warna merah, S *channel* dari putih ke merah

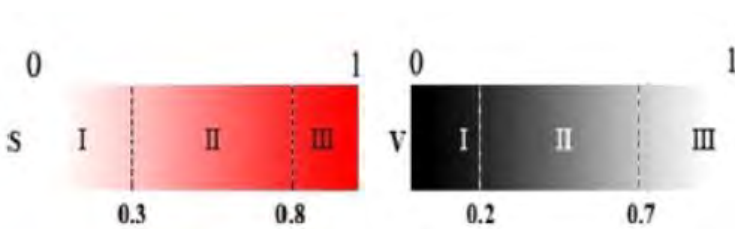
dengan *saturation* berbeda dengan *V channel* dari hitam ke putih dengan *intensity* berbeda.



Gambar 2.11 Skema Kuantisasi *Hue Channel* [6]

$$H = \begin{cases} 0 & \text{if } h \in (342, 16] \\ 1 & \text{if } h \in (16, 42] \\ 2 & \text{if } h \in (42, 64] \\ 3 & \text{if } h \in (64, 152] \\ 4 & \text{if } h \in (152, 195] \\ 5 & \text{if } h \in (195, 280] \\ 6 & \text{if } h \in (280, 342] \end{cases} \quad (2.5)$$

Pada saat nilai *S* cukup besar, untuk *instance S* lebih baik dari 0.8, bagian III pada Gambar 2.12 dapat dianggap sebagai warna merah murni. Ketika nilai *V* cukup kecil, untuk *instance V* lebih kecil dari 0.2 bagian I pada Gambar 2.12 bisa dianggap sebagai area warna hitam murni. Jadi, 3 *non-uniform bins* diekspresikan dari 0 sampai 2 sudah cukup untuk mewakili informasi *saturation* dan *value*. Persamaan untuk kuantisasi *saturation* dan *value* ditunjukkan oleh Persamaan(2. 6)



Gambar 2.12 Skema Kuantisasi S&V Channel [6]

$$S = \begin{cases} 0 & \text{if } s \in [0, 0.3) \\ 1 & \text{if } s \in [0.3, 0.8) \\ 2 & \text{if } s \in [0.8, 1) \end{cases}$$

$$V = \begin{cases} 0 & \text{if } v \in [0, 0.2) \\ 1 & \text{if } v \in [0.2, 0.7) \\ 2 & \text{if } v \in [0.7, 1) \end{cases} \quad (2.6)$$

Menurut skema pada Gambar 2.11 dan Gambar 2.12, fitur 1 dimensi vektor dibangun oleh 3 nilai *channel* seperti Persamaan(2. 7)

$$L = Q_s Q_v H + Q_v S + V \quad (2.7)$$

Q_s dan Q_v : koefisien *quantization* dari *saturation* dan *intensity(value)*.

Seperti metode *quantization* yang sering digunakan, koefisien *quantization* ditentukan sebagai $Q_s=Q_v=3$ maka L dapat dihitung dengan Persamaan (2. 8)

$$L = 9H + 3S + V \quad (2.8)$$

Jadi, 3 *channel* (*hue*, *saturation*, dan *value*) dapat terdistribusi dalam 1-dimensi vektor L dan $L \in \{0,1, \dots, 62\}$. Karena hasil kuantisasi hanya 63 *bins*, maka kompleksitas komputasi akan menurun pesat. Dengan mempertimbangkan karakter *non-uniform* pada 3 *channel* tersebut, hasil kuantisasi lebih mirip dengan cara visual manusia.

2.3.5 Analisis Histogram

Analisis histogram adalah tahapan pemilihan puncak-puncak histogram pada T_o dan T_b yang sudah dikuantisasi ruang warna HSV, hasil dari analisis histogram adalah lokasi dari TS_p dan TS_n dari sebuah citra yang ditunjukkan oleh ilustrasi pada Gambar 2.13 dan Gambar 2.14.

Puncak pada histogram dapat menjadi informasi dari warna pada citra. Untuk citra berwarna, warna dominan dapat diidentifikasi dengan puncak-puncak pada *global histogram*. Pemilihan puncak histogram dilakukan untuk menurunkan jumlah warna pada citra yang sudah dikuantisasi ruang warna HSV dan mengabaikan warna-warna yang sedikit muncul. Maka warna-warna tersebut merupakan warna dominan pada T_o dan T_b . Asumsi dari [6] yang sudah diujikan pada 1000 citra, bahwa tidak lebih dari 3 atau 4 warna dominan yang penting mendeskripsikan bagian T_o dan T_b . Langkah dalam analisis histogram adalah sebagai berikut :

1. Hitung *global histogram* dari T_o yang sudah menjalani proses kuantisasi ruang warna HSV dengan Persamaan(2. 9)

$$H^o = \frac{Num(f(x,y) = l_i)}{Num(T_o)},$$

$$(x,y) \in T_o, l_i \in \{0,1, \dots, 62\} \quad (2. 9)$$

$Num(T_o)$ = Jumlah piksel pada T_o .

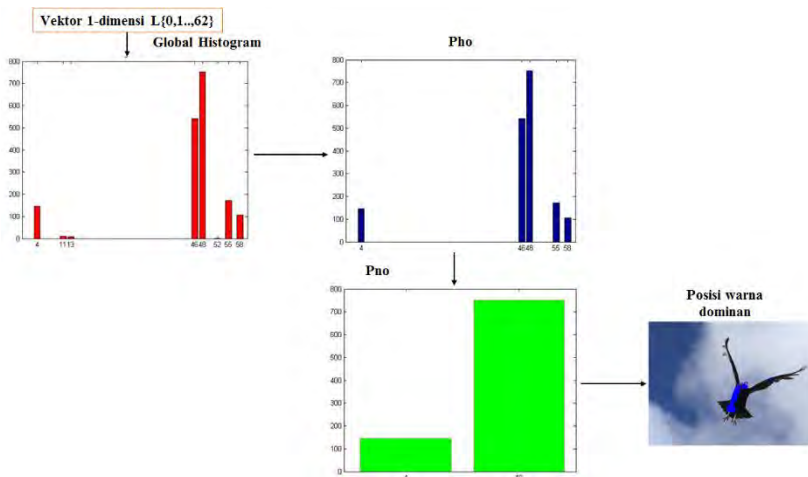
$Num(f(x,y))$ = Jumlah piksel dengan level warna l_i pada T_o .

2. Identifikasi semua puncak. $P_{ko} : P_{l_1}, P_{l_2}, \dots, P_{l_k}$, l_i adalah indeks dari puncak ke- i , dan $l_1 < l_2 < \dots < l_k$.
3. Hitung maksimum dan minimum puncak dari H^o . Hapus puncak terendah berdasarkan T_{ho} . Hitung T_{ho} dengan Persamaan (2. 10) dan puncak yang baru akan terbentuk $P_{ho} = P_{l_1}, P_{l_2}, \dots, P_{l_h}$

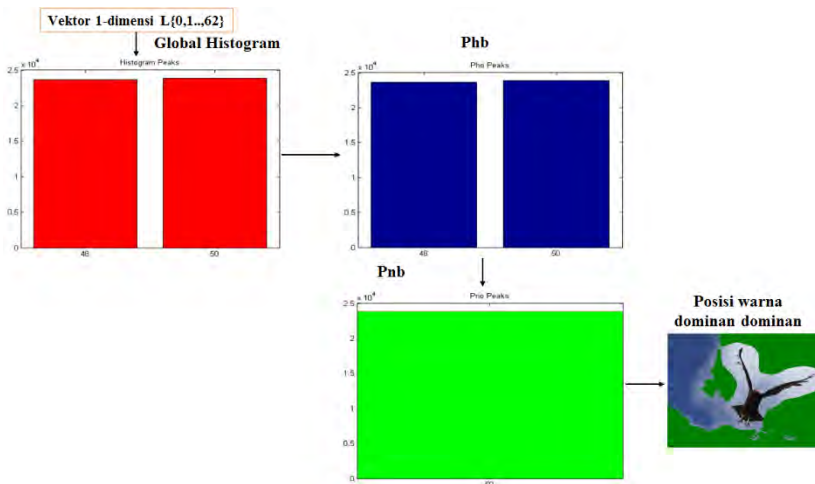
$$\begin{aligned}\mu_m &= \frac{P_{max} + P_{min}}{2} \\ \sigma_m &= \sqrt{\frac{\sum_{i=1}^k (P_{l_i} - \mu_m)^2}{k}} \\ T_{ho} &= \mu_m - \sigma_m\end{aligned}\tag{2. 10}$$

4. Hapus beberapa puncak menurut lebar threshold T_{wo} , $T_{wo} = 20$ ditentukan dengan asumsi bahwa tidak ada lagi warna dominan yang lebih dari 3 atau 4 pada objek To . Untuk 2 puncak yang berdekatan P_{l_i} dan P_{l_j} , jika $(l_j - l_i) < T_{wo}$ maka simpan puncak dengan nilai paling bagus dan hapus puncak lain dari P_{ho} .
5. Hasilnya adalah urutan puncak P_{no} dan warna dominan dari To yang ditentukan sebagai $C_o : l_1, l_2, \dots, l_n$.

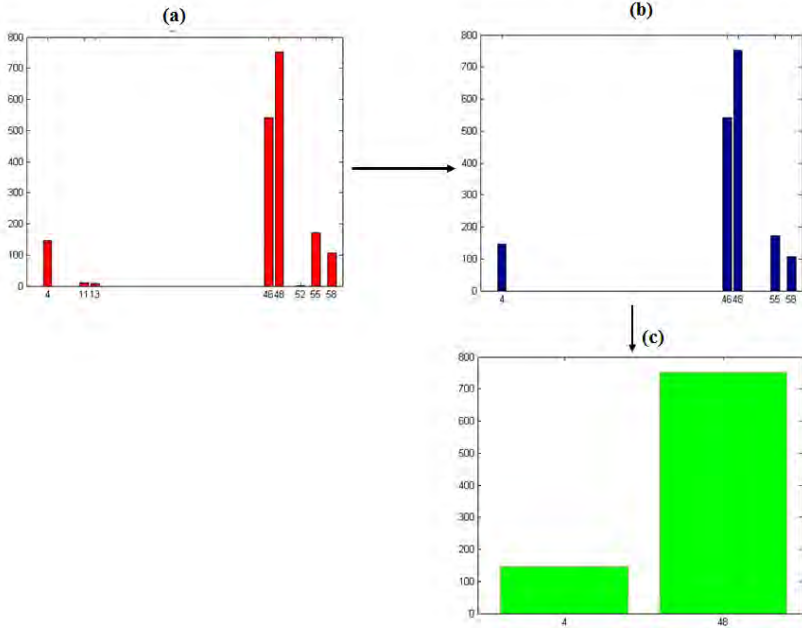
H^o pada Persamaan(2. 9) adalah *global histogram* dari bagian T_o yang sudah dikuantisasi ruang warna HSV sebelumnya yaitu 63 *bins*. Histogram dari bagian T_o pada sebuah contoh citra dapat ditunjukkan pada Gambar 2.15(a). P_{ho} pada langkah 3 adalah puncak histogram berdasarkan T_{ho} yang sudah dihitung dengan Persamaan (2. 10) P_{ho} dapat ditunjukkan pada Gambar 2.15(b). P_{no} adalah puncak berdasarkan T_{wo} yaitu dengan nilai 20 berdasarkan asumsi bahwa tidak lebih dari 3 warna dominan yang penting pada bagian T_o atau T_b . P_{no} ditunjukkan oleh Gambar 2.15(c).



Gambar 2.13 Ilustrasi Analisis Histogram (Bagian T_o)



Gambar 2.14 Ilustrasi Analisis Histogram (Bagian T_b)



Gambar 2.15 (a) Global histogram (b) Pho (c) Pno

Untuk T_b juga dilakukan seperti langkah-langkah diatas. Setelah warna dominan telah terpilih, maka SVM *training dataset* yang terdiri dari TS_p (*positive training set*) dan TS_n (*negative training set*) dapat dibentuk dengan Persamaan (2. 11). TS_p dan TS_n yang dihasilkan adalah dalam bentuk posisi x,y dari warna dominan yang dihasilkan oleh analisis histogram. Posisi tersebut bisa dilihat pada Gambar 2.16(a) yaitu titik-titik biru pada citra menunjukkan posisi TS_p . Pada Gambar 2.16(b) titik-titik hijau menunjukkan posisi TS_n .

$$\begin{aligned}
 TS_p &= \{(x, y) | f(x, y) = i, (x, y) \in T_o, \in C_o \\
 TS_n &= \{(x, y) | f(x, y) = i, (x, y) \in T_b, \in C_b
 \end{aligned}
 \quad (2. 11)$$

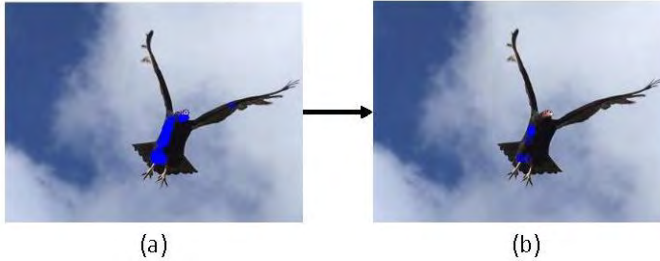


Gambar 2.16 (a) Posisi TS_p (b) Posisi TS_n

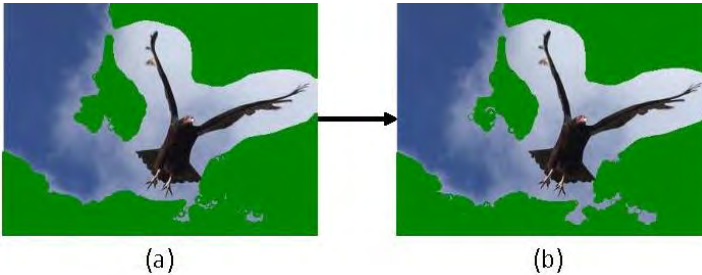
2.3.6 *Local Homogeneity Threshold*

Umumnya jumlah dari SVM *training dataset* terlalu besar untuk digunakan sebagai *training* secara langsung. *Neighborhood homogeneity threshold* digunakan untuk memilih piksel-piksel pada TS_p dan TS_n sebagai *training sample* pada SVM *training*. Untuk piksel $p(i, j)$ pada *training set* TS_p dan TS_n *local homogeneity* pada $n \times n$ *neighborhood* dihitung dengan Persamaan (2. 12). M_p adalah *local homogeneity* dengan $n \times n = 3$, $d(p, q)$ adalah *euclidean color distance* antara piksel p dan q pada ruang warna HSV yang sudah dikuantisasi. Jika $M_p \leq T_{lh}$ maka piksel tersebut termasuk dalam TS_p dan TS_n . Nilai T_{lh} ditentukan oleh pengguna, semakin baik nilai T_{lh} yang dimasukkan semakin tepat TS_p dan TS_n yang dihasilkan. Hasil *local homogeneity threshold* adalah posisi TS_p dan TS_n pada sebuah citra ditunjukkan pada Gambar 2.17 dan Gambar 2.18.

$$M_p = D_p^{n \times n} = \sum_{q \in N_p^{n \times n}} d(p, q) \quad (2. 12)$$



Gambar 2.17 TS_p Setelah Dilakukan *Local Homogeneity Threshold*



Gambar 2.18 TS_n Setelah Dilakukan *Local Homogeneity Threshold*

2.4 Ekstraksi Fitur

Setelah *training* piksel sudah terpilih maka dilakukan ekstraksi fitur sebagai vektor masukan pada SVM *classifier*. Ada 9 fitur yang akan diekstraksi sebagai vektor masukan yaitu 4 nilai warna, 2 nilai spasial, 2 nilai tekstur dan 1 nilai *saliency*.

2.4.1 Fitur Warna

Ekstraksi fitur warna digunakan untuk mengambil informasi warna pada tiap piksel pada citra, baik pada *training sample* maupun citra yang akan dilakukan segmentasi. Informasi warna yang diambil

adalah nilai r , g , dan b dari citra dengan ruang warna RGB dan nilai *intensity* (i) dalam citra yang sudah dikuantisasi ruang warna HSV.

2.4.2 Fitur *Saliency*

Ekstraksi fitur *saliency* digunakan untuk mengambil informasi *saliency* pada tiap piksel pada citra, baik pada *training sample* maupun citra yang akan dilakukan segmentasi. Informasi yang diambil adalah nilai *saliency* pada citra dengan *saliency map*.

2.4.3 Fitur Spasial

Ekstraksi fitur spasial digunakan untuk mengambil informasi posisi x, y pada tiap piksel pada citra, baik pada *training sample* maupun citra yang akan dilakukan segmentasi.

2.4.4 Fitur Tekstur

Ekstraksi fitur tekstur digunakan untuk mengambil informasi tekstur pada tiap piksel pada citra, baik pada *training sample* maupun citra yang akan dilakukan segmentasi. Informasi yang diambil adalah *local energy* dan *local gradient*. Untuk mendapatkan fitur tekstur pada tiap piksel diterapkan *steerable filter*. *Steerable filter* adalah kelas *filter* yang menyaring citra secara sembarang putaran. *Steerable filter* termasuk salah satu jenis *oriented filter*.

Langkah-langkah yang dilakukan untuk mengekstraksi fitur tekstur dengan *filter* ini adalah sebagai berikut :

1. Konversi citra RGB ke ruang warna YCbCr. Aplikasikan *steerable filter* pada komponen *luminance* (Y) [7].

$$Y = 0.299R + 0.587G + 0.114B \quad (2.13)$$

2. Memilih *basis filter* yang akan digunakan. Salah satu *basis filter* yang bisa digunakan adalah turunan pertama dari fungsi

Gaussian. Fungsi Gaussian 2 dimensi yang digunakan untuk pemrosesan citra jika ditulis dalam koordinat Cartesian x dan y adalah sebagai berikut :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.14)$$

Dengan nilai σ menunjukkan seberapa lebar *filter* yang akan digunakan. Pada koordinat x dan y , turunan pertama fungsi Gaussian dapat didekati pada sumbu x yaitu $G_1^{0^\circ}$ dan pada sumbu y yaitu $G_1^{90^\circ}$ seperti Persamaan (2.15).

$$\begin{aligned} G_1^{0^\circ} &= -\frac{x}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} \\ G_1^{90^\circ} &= -\frac{y}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} \end{aligned} \quad (2.15)$$

Matriks yang berisi nilai dari Persamaan (2.15) memiliki ukuran bergantung besar σ . Untuk σ bernilai 5, ukuran matriks $G_1^{0^\circ}$ dan $G_1^{90^\circ}$ adalah 7×7 . Nilai matriks tersebut ditunjukkan oleh Gambar 2.19

Dalam memilih *basis filter* yang akan digunakan, perlu diperhatikan apakah fungsi tersebut dapat ditulis dalam bentuk polinomial x dan polinomial y . Tidak semua fungsi dapat digunakan sebagai *basis filter* dalam *steerable filter*. Apabila suatu fungsi telah memenuhi syarat, berarti fungsi tersebut bersifat *separable* atau dapat diuraikan [8].

3. Menentukan fungsi interpolasi untuk mengatur orientasi dari *basis filter* yang dipilih. Untuk *basis filter* turunan pertama fungsi Gaussian, fungsi interpolasinya dengan Persamaan (2.16)

$$G_1^\theta = \cos \theta G_1^{0^\circ} + \sin \theta G_1^{90^\circ} \quad (2.16)$$

4. Menentukan sudut yang ingin digunakan pada *basis filter*.
Sudut yang digunakan adalah (0° , 30° , 60° , 90° , 120° , dan 150°) [7].
5. Hasil konvolusi pada citra I oleh *basis filter* berupa turunan pertama fungsi Gaussian didapat dari Persamaan(2. 17)

$$S = G_1^\theta * I = \cos \theta (G_1^{0^\circ} * I) + \sin \theta (G_1^{90^\circ} * I)$$

$$I * h[x, y] = \sum_{k,l} h[x - k, y - l]I[k, l] \quad (2. 17)$$

6. Dari hasil *steerable filter* pada citra dengan 6 orientasi yang ditunjukkan pada Gambar 2.20(a)-(f) , diambil nilai paling maksimum pada setiap piksel dalam nilai absolut yaitu menunjukkan nilai *local energy* pada fitur tekstur. $S_{x,y}^1$, $S_{x,y}^2$, $S_{x,y}^3$, $S_{x,y}^4$, $S_{x,y}^5$, dan $S_{x,y}^6$ adalah hasil *steerable filter* dari 6 orientasi pada lokasi piksel (x,y). $E_{x,y}$ adalah maksimum (dalam nilai absolut) dari 6 koefisien pada (x,y). Local energy ditunjukkan pada Gambar 2.20(g)

$$E_{x,y} = \max\{|S_{x,y}^1|, |S_{x,y}^2|, |S_{x,y}^3|, |S_{x,y}^4|, |S_{x,y}^5|, |S_{x,y}^6|\} \quad (2. 18)$$

7. Menghitung *gradient* dari hasil *steerable filter* dengan 6 orientasi dengan Persamaan(2. 19) dengan hasil ditunjukkan oleh Gambar 2.20(h).

$$\Delta x = S_{x+1,y}^l - S_{x,y}^l$$

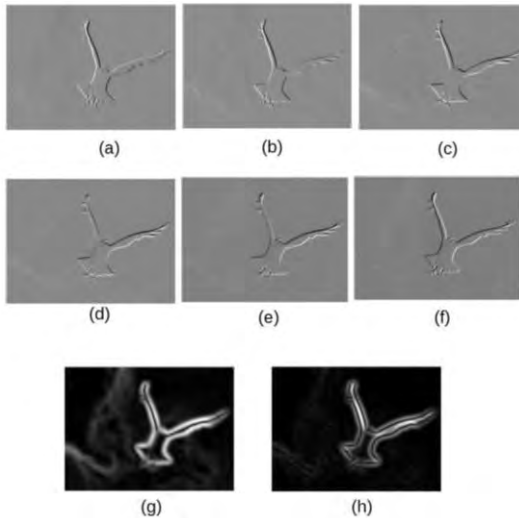
$$\Delta y = S_{x,y+1}^l - S_{x,y}^l \quad (l = 1,2,3,4,5,6)$$

$$G_{x,y}^l = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

$$G_{x,y} = \max\{G_{x,y}^l\} \quad (l = 1,2,3,4,5,6) \quad (2. 19)$$

$$\begin{aligned}
 G1_0 &= \begin{bmatrix} 0.016 & 0.014 & 0.009 & 0.000 & -0.009 & -0.014 & -0.016 \\ 0.022 & 0.019 & 0.011 & 0.000 & -0.011 & -0.019 & -0.022 \\ 0.025 & 0.022 & 0.013 & 0.000 & -0.013 & -0.022 & -0.025 \\ 0.027 & 0.024 & 0.014 & 0.000 & -0.014 & -0.024 & -0.027 \\ 0.025 & 0.022 & 0.013 & 0.000 & -0.013 & -0.022 & -0.025 \\ 0.022 & 0.019 & 0.011 & 0.000 & -0.011 & -0.019 & -0.022 \\ 0.016 & 0.014 & 0.009 & 0.000 & -0.009 & -0.014 & -0.016 \end{bmatrix} \\
 G1_90 &= \begin{bmatrix} 0.016 & 0.022 & 0.025 & 0.027 & 0.025 & 0.022 & 0.016 \\ 0.014 & 0.019 & 0.022 & 0.024 & 0.022 & 0.019 & 0.014 \\ 0.009 & 0.011 & 0.013 & 0.014 & 0.013 & 0.011 & 0.009 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.009 & -0.011 & -0.013 & -0.014 & -0.013 & -0.011 & -0.009 \\ -0.014 & -0.019 & -0.022 & -0.024 & -0.022 & -0.019 & -0.014 \\ -0.016 & -0.022 & -0.025 & -0.027 & -0.025 & -0.022 & -0.016 \end{bmatrix}
 \end{aligned}$$

Gambar 2.19 Matriks Turunan Gaussian dengan $\sigma=5$ [8]



Gambar 2.20 Fitur Tekstur (a) Orientasi 0 (b) Orientasi 30 (c) Orientasi 60 (d) Orientasi 90 (e) Orientasi 120 (f) Orientasi 150 (g) *local energy* (h) *local gradient*

2.5 Klasifikasi Piksel dengan *Support Vector Machine*

Setelah nilai fitur dari masing-masing piksel sudah di ekstraksi. Untuk menghasilkan segmentasi citra, nilai-nilai tersebut diklasifikasi dengan menggunakan *Support Vector Machine*. Di mana pada tahap ini akan dilakukan normalisasi data, *training SVM*, *testing SVM* dan evaluasi.

2.5.1 Normalisasi Data

Normalisasi adalah teknik *preprocessing* untuk melakukan penskalaan dan penstandaran data. Normalisasi yang dilakukan adalah dengan cara normalisasi skala, yaitu melakukan penskalaan data pada rentang tertentu, Di mana rentang yang umum digunakan adalah rentang 0-1. Normalisasi ini bertujuan untuk menghindari rentang data yang terlalu jauh. Untuk menghitung normalisasi skala menggunakan Persamaan (2. 20)

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2. 20)$$

Di mana Y adalah data hasil normalisasi, X adalah data sebelum di normalisasi X_{min} adalah nilai minimum yaitu 0 dan X_{max} adalah nilai maksimum yaitu 1.

2.5.2 Klasifikasi dengan SVM

SVM termasuk dalam *supervised learning*. *Supervised learning* merupakan metode yang membutuhkan *training* dan *testing*. Metode SVM berusaha mencari suatu fungsi pemisah yang memiliki generalisasi yang baik. Generalisasi yaitu kemampuan sebuah hipotesis untuk mengklasifikasikan kumpulan *training* yang terletak di tengah-tengah antara dua kelas. Mencari *hyperplane* yang optimal ekuivalen dengan memaksimalkan margin atau jarak antara bidang pembatas pada kedua kelas [1].

Pada bidang yang terbentuk dari pola data, SVM akan membentuk *hyperplane* yang memisahkan 2 *class*. Bentuk yang paling sederhana pada SVM adalah linier SVM. Sebagai acuan, terdapat sebuah *training* set D yang dapat dinotasikan dengan Persamaan (2. 21)

$$D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1, 1\}\}_{i=1}^n \quad (2. 21)$$

Nilai $y_i \in \{-1, 1\}$ merupakan penunjuk *class* atau label bernilai 1 atau -1 dari kumpulan data x_i . *Hyperplane* yang terbaik merupakan *hyperplane* yang memisahkan data dengan $y_i = 1$ dan $y_i = -1$. *Hyperplane* yang memisahkan tersebut dapat dinotasikan dengan Persamaan(2. 22)

$$w \cdot x - b = 0 \quad (2. 22)$$

Titik data x_i yang termasuk kelas 1 dapat dinotasikan sebagai titik yang memenuhi Persamaan(2. 23) dan titik data x_i yang termasuk kelas -1 dapat dinotasikan dengan Persamaan(2. 24)

$$w \cdot x - b \geq 1, y_i = 1 \quad (2. 23)$$

$$w \cdot x - b \leq -1, y_i = -1 \quad (2. 24)$$

Hyperplane yang ideal adalah *hyperplane* yang mempunyai margin maksimal dalam memisahkan kelas-kelas data. Sesuai dengan penerapan rumus perhitungan jarak antara titik dengan garis, maka jarak antara data terdekat dengan batas tengah *hyperplane* adalah $\frac{1}{||w||}$. Untuk memaksimalkan margin, maka nilai $||w||$ harus diminimalisasi. Peminimalisasian nilai $||w||$ sama artinya dengan menyelesaikan permasalahan *quadratic programming*. Permasalahan *quadratic programming* ini akan mencari titik minimal Persamaan (2. 25) dengan memperhatikan batasan dari Persamaan (2. 26)

$$f: \frac{1}{2} ||w||^2 \quad (2.25)$$

$$y_i(x_i \cdot w + b) - 1 \geq 0, \forall_i \quad (2.26)$$

Permasalahan *quadratic programming* ini dapat dipecahkan dengan berbagai teknik komputasi. Salah satu metode pemecahannya adalah dengan menggunakan *Lagrange Multiplier*. *Lagrange Multiplier* menggunakan variable α sehingga persamaan akan lebih mudah dihitung. Variabel α bernilai nol atau positif untuk Persamaan (2.27)

$$L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^l \alpha_i (y_i(x_i \cdot w + b) - 1) \quad (2.27)$$

Nilai optimal dari Persamaan (2.29) dapat dihitung dengan meminimalkan L terhadap w dan b , dan memaksimalkan L terhadap α_i . Persamaan (2.27) juga dapat dimodifikasi sehingga hanya mengandung α_i dengan memperhatikan nilai $L = 0$ pada titik optimal *gradient* menjadi Persamaan (2.28) dengan batasan Persamaan (2.29). Data yang berkorelasi dengan α_i yang positif inilah yang disebut dengan *support vector*.

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j x_i x_j \quad (2.28)$$

$$\alpha_i \geq 0 \text{ dan } \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.29)$$

SVM pada dasarnya adalah sebuah model klasifikasi linear, namun bisa digunakan untuk melakukan klasifikasi non-linier dengan menggunakan *kernel trick* seperti Persamaan (2.30).

$$K(x_i x_j) = \phi(x_i) \cdot \phi(x_j) \quad (2.30)$$

Kernel trick dalam SVM adalah cara untuk melakukan klasifikasi data non-linear dengan cara mentransformasikan ruang data asli menjadi ruang data dengan dimensi yang lebih tinggi. *Kernel* yang biasa digunakan dalam SVM antara lain ditunjukkan Tabel 2.1.

Tabel 2.1 Fungsi Kernel yang Biasa Digunakan dalam SVM

Nama Kernel	Inner Product Kernel
Linear	$x^T x$
Polinomial	$(x^T x_i + 1)^p$ dengan nilai p bebas, ditentukan oleh pengguna.
Radial-basis function (RBF/Gaussian)	$\exp\left(-\frac{\ x-x'\ _2^2}{2\sigma^2}\right)$ dengan nilai σ bebas, ditentukan oleh pengguna.

2.5.3 Metode Evaluasi

Evaluasi dilakukan dengan menghitung *error rate* dan akurasi dari perbandingan citra hasil segmentasi dengan *ground truth* menghitung *error rate* dan akurasi ditunjukkan oleh Persamaan (2. 31)

$$ER = \frac{N_f + N_m}{N_t} \times 100\%$$

$$Akurasi = 100 - ER \quad (2. 31)$$

Di mana variabel N_f menunjukkan jumlah *false-segmented* pada piksel citra, N_m menunjukkan jumlah *miss-segmented* pada piksel citra dan N_t adalah total piksel citra.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan perancangan program yang dibuat. Perancangan akan dibagi menjadi tiga proses utama, yaitu:

1. Pemilihan training *piksel* dengan *visual saliency* berbasis SVM.
2. Ekstraksi fitur.
3. Klasifikasi piksel menggunakan *Support Vector Machine*.

Pada bab ini akan dijelaskan gambaran umum setiap program utama dalam *flowchart* selanjutnya untuk penjelasan lebih detail akan disajikan dalam *pseudocode* dengan variabel yang digunakan ditunjukkan pada Tabel 3.1 dan Tabel 3.2.

**Tabel 3.1 Variabel yang Digunakan pada Perancangan
(Bagian 1)**

No.	Nama Data	Tipe Data	Keterangan
1.	<i>Image</i>	<i>Uint8</i>	Cita masukan (RGB)
2.	<i>SaliencyMap</i>	<i>Double</i>	Hasil <i>visual saliency detection</i>
3.	<i>Ro</i>	<i>Logical</i>	Citra biner (perkiraan objek)
4.	<i>Rb</i>	<i>Logical</i>	Citra biner (perkiraan <i>background</i>)
5.	<i>Mo</i>	<i>Logical</i>	Hasil erosi dari <i>Ro</i>
6.	<i>Mb</i>	<i>Logical</i>	Hasil dilasi dari <i>Ro</i>
7.	<i>To</i>	<i>Uint8</i>	<i>Mo</i> (RGB)

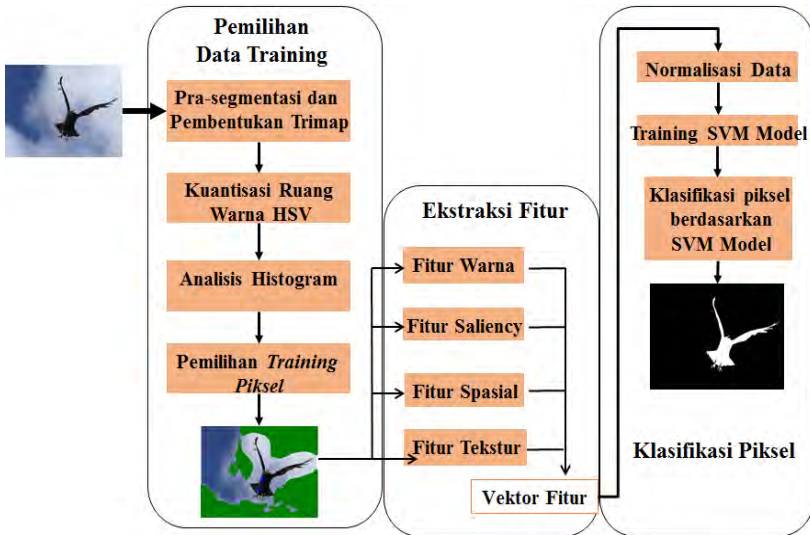
Tabel 3.2 Variabel yang Digunakan pada Perancangan (Bagian 2)

No.	Nama Data	Tipe Data	Keterangan
8.	<i>Tb</i>	<i>Uint8</i>	<i>Mb</i> RGB)
9.	<i>LTo</i>	<i>Double</i>	Vektor 1-dimensi (hasil kuantisasi HSV) pada <i>To</i>
10.	<i>LTb</i>	<i>Double</i>	Vektor 1-dimensi (hasil kuantisasi HSV) pada <i>Tb</i>
11.	<i>Pno, Pnb</i>	<i>int</i>	Urutan warna dominan
12.	<i>TSpX</i>	<i>Double</i>	Posisi x dari <i>TSp</i>
13.	<i>TSpY</i>	<i>Double</i>	Posisi y dari <i>TSp</i>
14.	<i>TSnX</i>	<i>Double</i>	Posisi x dari <i>TSn</i>
15.	<i>TSnY</i>	<i>Double</i>	Posisi y dari <i>TSn</i>

3.1 Perancangan Secara Umum

Secara umum proses segmentasi citra menggunakan Saliency-SVM tidak jauh berbeda dengan proses-proses segmentasi citra berbasis SVM lainnya. Pada Saliency-SVM tahap pertama adalah pemilihan *training* piksel menggunakan *visual saliency* berbasis SVM yang di dalamnya terdapat proses pra-segmentasi citra dan pembentukan *trimap*, kuantisasi ruang warna HSV, analisis histogram serta pemilihan *training* piksel dengan *local homogeneity threshold* yang akan menghasilkan *training sample* (TS_p dan TS_n), selanjutnya dilakukan ekstraksi fitur dari *training sample* dan citra masukan, setelah nilai fitur sudah diekstraksi nilai tersebut dinormalisasi dengan

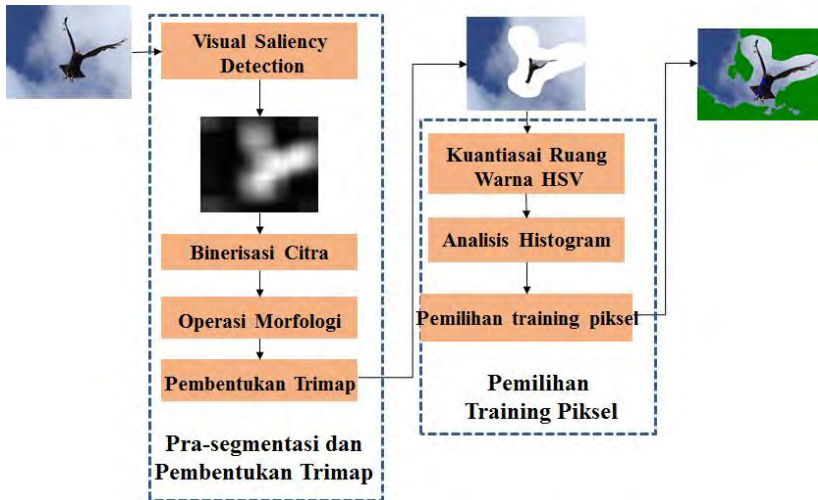
normalisasi skala, lalu dilakukan pembelajaran *training sample* dengan SVM untuk menghasilkan model dan terakhir klasifikasi piksel dengan SVM berdasarkan model yang dihasilkan sebelumnya seperti ditunjukkan Gambar 3.1.



Gambar 3.1 Diagram Alir Program Secara Umum

3.2 Perancangan Pemilihan *Training* Piksel dengan *Visual Saliency* berbasis SVM

Pada segmentasi citra dengan menggunakan SVM dibutuhkan *training sample* untuk proses *training* pada SVM yang nantinya akan menghasilkan SVM Model. Pemilihan *training* piksel dengan *visual saliency* berbasis SVM ini akan melalui tahapan pra-segmentasi dan pembentukan *trimap*, kuantisasi ruang warna HSV, analisis histogram dan pemilihan *training sample* dengan *local homogeneity threshold* seperti pada Gambar 3.2.



Gambar 3.2 Diagram Alir Pemilihan *Training* Piksel dengan *Visual Saliency* Berbasis SVM

3.2.1 Perancangan Pra-segmentasi dan Pembentukan *Trimap*

Pada pemilihan training piksel terdapat proses pra-segmentasi dan pembentukan *trimap* seperti yang sudah ditunjukkan Gambar 3.2 proses tersebut bertujuan untuk memperkirakan lokasi objek dan *background* pada citra masukan. Proses-proses yang terdapat di dalam pra-segmentasi dan pembentukan *trimap* akan dijelaskan oleh *pseudocode* pada Gambar 3.3.

3.2.2 Kuantisasi Ruang Warna HSV

Kuantisasi ruang warna HSV dilakukan dengan tujuan untuk menurunkan kompleksitas pada saat melakukan analisis histogram pada T_o dan T_b . Hasil dari kuantisasi ruang warna HSV adalah vektor 1-dimensi dengan *63-bins*. Implementasi pada kuantisasi ruang warna HSV menggunakan *pseudocode* pada Gambar 3.4.

Masukan	Citra Masukan (RGB)
Keluaran	<i>Trimap</i> (To dan Tb)
1.	Image \leftarrow konversi ke gray level
2.	Image \leftarrow Resize menjadi 64 x 64 piksel
3.	SaliencyMap \leftarrow Spectral residual dengan menggunakan Persamaan (2.1)
4.	SaliencyMap \leftarrow Kembalikan ke ukuran citra awal
5.	Ro & Rb \leftarrow binerisasi (SaliencyMap)
6.	Mo \leftarrow erosi sesuai Persamaan (2.3)
7.	Mb \leftarrow dilasi sesuai Persamaan (2.4)
8.	To \leftarrow citra dengan posisi Mo pada citra masukan (RGB)
9.	Tb \leftarrow citra dengan posisi Mb pada citra masukan (RGB)

Gambar 3.3 Pseudocode Pra-segmentasi dan Pembentukan *trimap*

Masukan	To dan Tb
Keluaran	LTo dan LTb (hasil kuantisasi)
1.	Konversi To dan Tb ke ruang warna HSV $H[0,360], S[0,1], V[0,1]$
2.	Kuantisasi H dengan menggunakan Persamaan (2.5)
3.	Kuantisasi S dan V dengan menggunakan Persamaan (2.6)
4.	Kuantisasi H, S dan V dengan Persamaan (2.8)

Gambar 3.4 Pseudocode Kuantisasi Ruang Warna HSV

3.2.3 Perancangan Analisis Histogram

Setelah bagian To dan Tb sudah dikuantisasi, untuk mencari warna dominan pada To dan Tb dilakukan analisis histogram dengan menggunakan *pseudocode* pada Gambar 3.5.

Di mana analisis histogram melakukan pemilihan puncak-puncak histogram untuk memilih warna dominan pada To dan Tb .

Warna dominan adalah salah satu ciri pembeda yang sudah dijelaskan pada bab 2.3.

Setelah warna dominan sudah terpilih, maka TS_p dan TS_n dapat dibentuk dengan Persamaan (2. 11). Hasil pembentukan TS_p dan TS_n adalah posisi x,y dari warna dominan tersebut pada To dan Tb .

Masukan	LTo dan LTb (hasil kuantisasi)
Keluaran	Co dan Cb (Warna dominan pada To dan Tb)
1.	Hitung <i>global histogram</i> dengan menggunakan Persamaan (2. 9)
2.	Identifikasi seluruh puncak histogram.
3.	Hapus puncak berdasarkan T_{ho} yang dihitung dari Persamaan (2. 10) jika puncak kurang dari T_{ho} maka puncak tersebut dihapus.
4.	Hapus puncak berdasarkan lebar threshold T_{wo} , $T_{wo} = 20$, Untuk 2 puncak yang berdekatan P_{li} dan P_{lj} , jika $(l_j - l_i) < T_{wo}$ maka simpan puncak dengan nilai paling bagus dan hapus puncak lain dari P_{ho} .
5.	Hasilnya adalah urutan puncak P_{no} dan P_{nb} , serta warna dominan Co dan Cb

Gambar 3.5 Pseudocode Analisis Histogram

3.2.4 Perancangan Pemilihan *Training* Pikel dengan *Local Homogeneity Threshold*

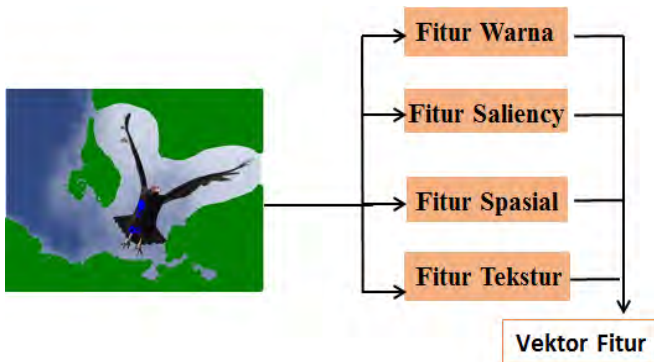
Masukan	Posisi x,y (TS_p dan TS_n)
Keluaran	Posisi x,y (TS_p dan TS_n , yang baru)
1.	Hitung euclidean distance dari piksel dengan Persamaan (2. 12)
2.	Cek jika $M_p \leq T_{lh}$ Maka posisi piksel pada M_p termasuk training sample (TS_p, TS_n)

Gambar 3.6 Pseudocode Local Homogeneity Threshold

Setelah posisi x, y warna dominan pada To dan Tb atau bisa disebut dengan TS_p dan TS_n sudah dibentuk maka dilakukan pemilihan *training sample* dengan *Local Homogeneity Threshold* dengan *pseudocode* pada Gambar 3.6.

3.3 Perancangan Ekstraksi Fitur

Proses utama lainnya pada Saliency-SVM ini adalah ekstraksi fitur yang nantinya akan dijadikan vektor masukan pada klasifikasi dengan *support vector machine*. Fitur yang diekstraksi adalah fitur warna, fitur *saliency*, fitur spasial, dan fitur tekstur. Secara umum tahapan pada ekstraksi fitur adalah seperti Gambar 3.7.



Gambar 3.7 : Diagram alir ekstraksi fitur secara umum

3.3.1 Perancangan Fitur Warna

Fitur warna adalah mengambil nilai warna (r , g , dan b) pada citra RGB dan nilai *intensity* (i) pada citra yang sudah dikuantisasi HSV. Mengambil nilai fitur warna menggunakan *pseudocode* yang ditunjukkan oleh Gambar 3.8. Di mana mengambil nilai warna dari posisi pada piksel. Untuk vektor *training*, nilai yang diambil berdasarkan posisi *training* piksel yang sudah dihasilkan pada proses sebelumnya. Untuk vektor fitur *testing*, nilai yang diambil berdasarkan seluruh posisi piksel pada citra masukan.

3.3.2 Perancangan Fitur *Saliency*

Fitur *saliency* adalah mengambil nilai piksel pada citra *Saliency Map* yang sudah dibentuk dengan *visual saliency detection*. Untuk vektor *training*, nilai yang diambil berdasarkan posisi *training* piksel yang sudah dihasilkan pada proses sebelumnya. Untuk vektor fitur *testing*, nilai yang diambil berdasarkan seluruh posisi piksel pada citra masukan. Mengambil nilai fitur *saliency* menggunakan *pseudocode* pada Gambar 3.9.

3.3.3 Perancangan Fitur Spasial

Fitur spasial adalah nilai posisi x,y dari citra baik itu citra *testing* maupun *training*. Karena TS_p dan TS_n yang dihasilkan sudah dalam bentuk spasial atau posisi x,y maka nilai x,y pada *training* langsung dapat diekstraksi dari hasil *local homogeneity threshold*. Sedangkan untuk *testing* diambil dari seluruh posisi (x,y) pada citra masukan, seperti *pseudocode* pada Gambar 3.10.

3.3.4 Perancangan Fitur Tekstur

Fitur tekstur adalah mengambil nilai *local energy* dan nilai *local gradient* yang sudah dijelaskan pada bab 2.4.4. Di mana nilai-nilai tekstur diambil dari citra yang sudah diaplikasikan *steerable filter*. Untuk mengambil nilai fitur tekstur menggunakan *pseudocode* pada Gambar 3.11.

Masukan	Image, dan posisi x,y dari image
Keluaran	Nilai warna
1. for $i \leftarrow \text{size}(\text{image})$	
2. $r \leftarrow \text{image}(i), r$	
3. $g \leftarrow \text{image}(i), g$	
4. $b \leftarrow \text{image}(i), b$	
5. $\text{Intensity} \leftarrow \text{image}(i), \text{intensity}$	
6. end	

Gambar 3.8 Pseudocode Ekstraksi Fitur Warna

Masukan	Image, posisi x,y dari image
Keluaran	Nilai saliency(S)
1.	for i \leftarrow size (image)
2.	S \leftarrow image(i), s
3.	end

Gambar 3.9 Pseudocode Ekstraksi Fitur Saliency

Masukan	Image
Keluaran	Nilai x,y(Testx, Testy)
1.	for i \leftarrow size(image,1)
2.	for j \leftarrow size (image,2)
3.	Testx \leftarrow i
4.	Testy \leftarrow j

Gambar 3.10 Pseudocode Ekstraksi Fitur Spasial

Masukan	Image, posis x,y citra
Keluaran	Nilai local energy dan local gradient
1.	Konversi citra RGB - YcbCr dengan Persamaan (2.13)
2.	Aplikasikan steerable filter yang sudah dijelaskan pada Persamaan (2.14)(2.17)(2.13)
3.	E \leftarrow hitung nilai local energy dengan Persamaan (2.18)
4.	G \leftarrow Hitung nilai local gradient dengan Persamaan (2.19)
7.	for i \leftarrow size (image)
8.	E \leftarrow image (i), E
9.	G \leftarrow image (i), G
10.	end

Gambar 3.11 Pseudocode Ekstraksi Fitur Tekstur

3.4 Perancangan Klasifikasi Piksel dengan SVM

Setelah pemilihan *training* piksel dan ekstraksi fitur, dilakukan klasifikasi piksel dengan SVM. Secara umum tahapan klasifikasi dengan SVM ditunjukkan Gambar 3.7

3.4.1 Perancangan Normalisasi Data

Normalisasi data dilakukan pada vektor fitur yang dihasilkan dari proses sebelumnya. Di mana normalisasi dilakukan dengan rentang nilai di antara $[0,1]$ normalisasi dilakukan dengan Persamaan (2. 20).

3.4.2 Perancangan *Training* SVM

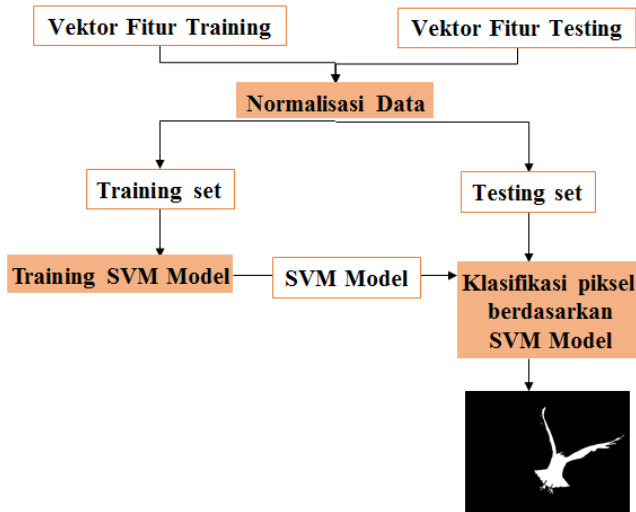
Proses *training* pada SVM ini adalah bertujuan untuk menghasilkan SVM Model yang nantinya akan dijadikan Model pada saat *testing* SVM. Data yang digunakan pada proses ini adalah data hasil pemilihan *training* piksel dengan *visual saliency* berbasis SVM dengan nilai-nilai fitur yang sudah diekstraksi. Langkah-langkah pada *training* SVM ditunjukkan oleh *pseudocode* pada Gambar 3.13.

3.4.3 Perancangan *Testing* SVM

Proses *testing* pada SVM adalah proses Di mana mengklasifikasi seluruh piksel yang terdapat pada citra masukan. *Testing* pada SVM dilakukan berdasarkan Model yang dihasilkan oleh proses *training* SVM. Data yang digunakan adalah seluruh piksel pada citra dengan nilai-nilai fitur yang sudah diekstraksi sebelumnya. Langkah-langkah pada *testing* SVM ditunjukkan oleh *pseudocode* pada Gambar 3.14.

3.4.4 Perancangan Evaluasi

Evaluasi yang dilakukan adalah membandingkan hasil segmentasi yang dilakukan oleh Saliency-SVM dengan *ground truth*. Evaluasi yang dilakukan dengan menghitung *error rate* dan akurasi sesuai dengan Persamaan (2. 31).



Gambar 3.12 Diagram Alir Klasifikasi Piksel dengan SVM Secara Umum

Masukan	Vektor fitur TS_p & TS_n , Kernel, C
Keluaran	Model SVM
1.	<code>datatrain ← vector fitur TS_p & TS_n</code>
2.	<code>dataclass ← class TS_p; TS_n</code>
3.	<code>Kernel ← fungsi kernel</code>
4.	<code>C ← input nilai C</code>
5.	<code>SVM Mode ← svmtrain(datatrain, dataclass, Kernel, C)</code>

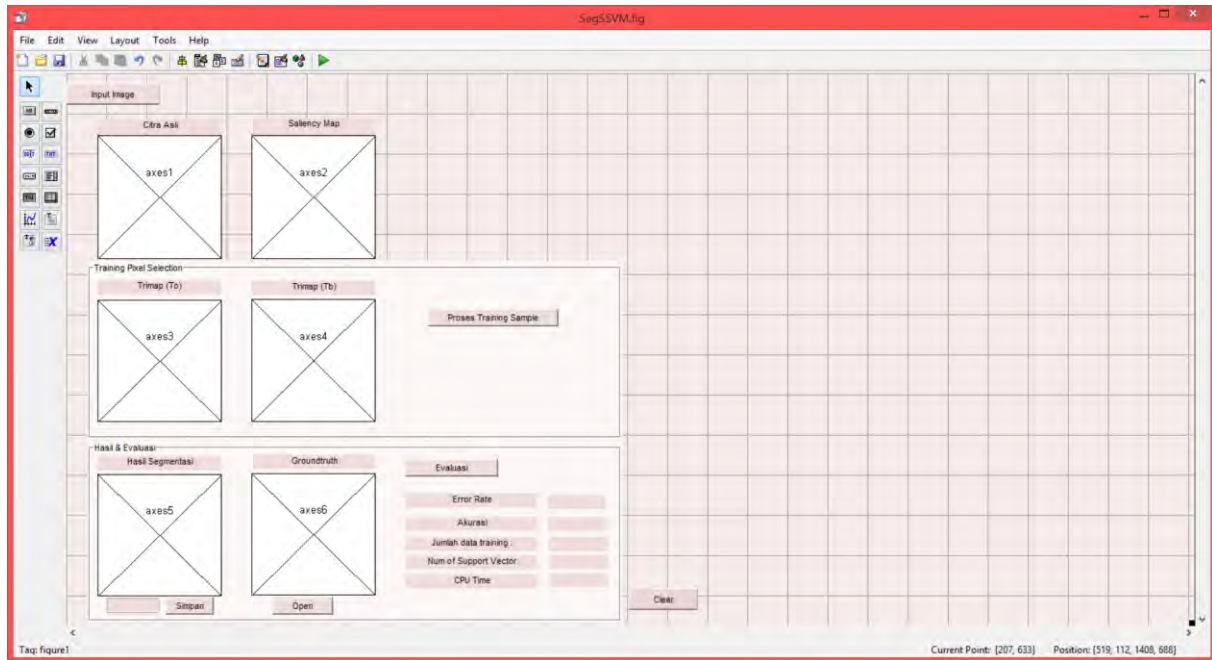
Gambar 3.13 Pseudocode Training SVM

Masukan	Vektor fitur citra masukan, SVM Model
Keluaran	Citra hasil segmentasi
1.	<code>datatest ← vektor fitur seluruh piksel citra masukan</code>
2.	<code>hasil ← svmclassify(Model, datatest)</code>
3.	<code>Hasil ← kembalikan ke posisi pada citra</code>

Gambar 3.14 Pseudocode Testing SVM

3.5 Perancangan Antarmuka Aplikasi

Dalam aplikasi ini terdapat satu jendela yang digunakan, di mana jendela tersebut adalah jendela utama aplikasi. Rancangan antarmuka dapat dilihat pada Gambar 3.15. Dalam jendela utama aplikasi ini ditampilkan citra asli (masukan dari pengguna) yang akan diolah, kemudian ditampilkan *saliency map* dari citra. Setelah itu ditampilkan *trimap* (T_o dan T_b). Terdapat 1 tombol untuk memproses *trimap* (T_o dan T_b) untuk pemilihan data *training*, ekstraksi fitur dan klasifikasi piksel. Lalu hasil segmentasi ditampilkan dan dapat disimpan oleh pengguna, lalu pengguna memasukkan *ground truth* untuk melakukan evaluasi pada hasil segmentasi dibandingkan dengan *ground truth*.



Gambar 3.15 Tampilan Rancangan Antarmuka Aplikasi

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah MATLAB 8.1.0 (R2013a) yang dipasang pada sistem operasi Windows 8.

4.2 Implementasi

Pada bagian ini akan dijelaskan implementasi dari perancangan perangkat lunak. Pada bagian implementasi akan dijelaskan implementasi secara umum, pemilihan *training* piksel dengan *visual saliency* berbasis SVM, ekstraksi fitur, dan klasifikasi piksel dengan SVM.

4.2.1 Implementasi Secara Umum

Pada bagian ini akan dijelaskan kode implementasi secara umum, di mana terdapat 3 proses utama yaitu pemilihan *training* piksel dengan *visual saliency* berbasis SVM, ekstraksi fitur, dan klasifikasi piksel dengan SVM. Implementasi yang dilakukan berdasarkan *pseudocode* dan diagram alir pada bab 3 dan persamaan-persamaan pada bab 2.

Secara umum implementasi ini berdasarkan Gambar 3.1, dalam matlab, kode implementasi ditunjukkan oleh Kode Sumber 4.1 dan Kode Sumber 4.2, di mana baris 1-28 merupakan salah satu proses utama yaitu pemilihan *training* piksel dengan *visual saliency* berbasis SVM.

1.	<code>%readfile</code>
2.	<code>RImg = imread(filename);</code>
3.	
4.	<code>SaliencyMap = SpectralResidualSaliency(RImg);</code>
5.	
6.	<code>level = graythresh(SaliencyMap);</code>
7.	<code>Ro = im2bw(SaliencyMap, level);</code>
8.	<code>se = strel('square',70);</code>
9.	<code>Mo = imerode(Ro, se);</code>
10.	<code>Mb = (imdilate(Ro, se)-Ro) + Rb;</code>
11.	
12.	<code>To = trimapgeneration(RImg, Mo);</code>
13.	
14.	<code>Tb = trimapgeneration(RImg, Mb);</code>
15.	
16.	<code>ToHsv = rgb2hsv (To);</code>
17.	<code>TbHsv = rgb2hsv (Tb);</code>
18.	<code>ImgHSV = rgb2hsv (RImg);</code>
19.	
20.	<code>LToHSV = KuantisasiHSV(ToHsv);</code>
21.	<code>LTbHSV = KuantisasiHSV(TbHsv);</code>
22.	<code>LImgHSV = KuantisasiHSV(ImgHSV);</code>
23.	
24.	<code>Pno = Hist_Peak_Finding(LToHSV, Mo);</code>
25.	<code>Pnb = Hist_Peak_Finding(LTbHSV, Mb);</code>
26.	
27.	<code>[TSpX,TSpY] = Homogeneity_Thresholding(XTo, YTo, LToHSV, Tlh);</code>
28.	<code>[TSnX, TSnY] = Homogeneity_Thresholding(XTb, YTb, LTbHSV, Tlh);</code>

Kode Sumber 4.1 Kode Implementasi Saliency-SVM Secara Umum (Bagian 1)

29.	<code>s = FiturSaliency(SaliencyMap, TSpX, TSpY);</code>
30.	
31.	<code>[r, g, b] = FiturWarna (RImg, TSpX,TSpY);</code>
32.	
33.	<code>intensity = FiturI(LToHSV, TSpX, TSpY);</code>
34.	
35.	<code>[localenergy, gradient] = FiturTekstur(RImg); [E, G] = ekstraksiFiturtekstur(localenergy, gradient, TSpX, TSpY);</code>
36.	
37.	<code>datatrain = scalingdata (trainlist, 0,1);</code>
38.	
39.	<code>SVMModel = svmtrain(datatrain, trainclass,'kernel_function','linear','boxc onstraint',0.1,'autoscale',false);</code>
40.	
41.	<code>SegmentasiCitra = svmclassify(SVMModel, datatest);</code>

Kode Sumber 4.2 Kode Implementasi Saliency-SVM Secara Umum (Bagian 2)

Pada baris 29-35 merupakan kode implementasi pada proses utama yaitu ekstraksi fitur. Baris 37-41 adalah proses utama klasifikasi piksel dengan SVM.

4.2.2 Implementasi Pemilihan *Training* Piksel dengan *Visual Saliency* Berbasis SVM

Pada bagian ini akan dijelaskan implementasi dari pemilihan *training* piksel dengan *visual saliency* berbasis SVM. Di mana di dalamnya terdapat pra-segmentasi dan pembentukan *trimap*, kuantisasi ruang warna HSV, analisis histogram dan *local homogeneity threshold*.

4.2.2.1 Implementasi Pra-segmentasi dan Pembentukan *Trimap*

Citra masukan awalnya dilakukan pra-segmentasi dengan tujuan menghasilkan *trimap* yang akan diproses untuk pemilihan *training* piksel. Implementasi ini merupakan implementasi dari Gambar 3.3. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.3. Di mana baris 1 untuk membaca citra yang dimasukkan oleh pengguna, baris 2 menunjukkan pembentukan *saliency map* dengan *spectral residual*, baris 3-4 menunjukkan binerisasi citra dengan metode otsu, baris 5-7 menunjukkan dilasi dan erosi citra dengan operasi morfologi, dan baris 8-9 merupakan pembentukan *trimap* berdasarkan hasil erosi dan dilasi.

1.	<code>RImg = imread(filename);</code>
2.	<code>SaliencyMap = SpectralResidualSaliency(RImg);</code>
3.	<code>level = graythresh(SaliencyMap);</code>
4.	<code>Ro = im2bw(SaliencyMap, level);</code>
5.	<code>se = strel('square', 70);</code>
6.	<code>Mo = imerode(Ro, se);</code>
7.	<code>Mb = (imdilate(Ro, se) - Ro) + Rb;</code>
8.	<code>To = trimapgeneration(RImg, Mo);</code>
9.	<code>Tb = trimapgeneration(RImg, Mb);</code>

Kode Sumber 4.3 Kode Implementasi Pra-segmentasi dan Pembentukan *Trimap*

Baris 2 adalah fungsi pembentukan *saliency map* dengan *spectral residual*, implementasi ini merupakan implementasi dari Persamaan (2. 1). Kode implementasi dari fungsi *spectral residual* ditunjukkan oleh Kode Sumber 4.4. Di mana implementasi ini melakukan analisis citra pada *domain* frekuensi, dengan menganalisis *log-spectrum* dari citra *saliency map* dibentuk berdasarkan *spectral residual* dari *amplitude spectrum* dari *fourier transform* pada citra.

1.	<code>myFFT = fft2(inImg);</code>
2.	<code>myLogAmplitude = log(abs(myFFT));</code>
3.	<code>myPhase = angle(myFFT);</code>
4.	<code>mySpectralResidual = myLogAmplitude - imfilter(myLogAmplitude, fspecial('average',3), 'replicate');</code>
5.	<code>saliencyMap = abs(ifft2(exp(mySpectralResidual + 1i*myPhase))).^2;</code>
6.	<code>saliencyMap = imfilter(saliencyMap, fspecial('gaussian',[20,20],8));</code>

Kode Sumber 4.4 Kode Implementasi Fungsi *Spectral Residual*

1.	<code>for i=1:size(Mo,1)</code>
2.	<code> for j=1:size(Mo,2)</code>
3.	<code> if Mo(i,j)==0</code>
4.	<code> To(i,j,1) = 255;</code>
5.	<code> To(i,j,2) = 255;</code>
6.	<code> To(i,j,3) = 255;</code>
7.	<code> else</code>
8.	<code> To(i,j,1) = RImg(i,j,1);</code>
9.	<code> To(i,j,2) = RImg(i,j,2);</code>
10.	<code> To(i,j,3) = RImg(i,j,3);</code>

Kode Sumber 4.5 Kode Implementasi Fungsi Pembentukan *Trimap*

Setelah *saliency map* dibentuk dengan fungsi *spectral residual*, maka dilakukan binerisasi dengan metode otsu yang ditunjukkan oleh baris 3-4 pada Kode Sumber 4.3. Lalu dilakukan operasi morfologi pada baris 5-7 untuk menghasilkan *mask* objek dan *mask background* yang nantinya dijadikan patokan lokasi pada pembentukan *trimap*. Fungsi pembentukan *trimap* ditunjukkan oleh Kode Sumber 4.5. Di mana membentuk *trimap* berdasarkan posisi *mask* objek dan *mask background* dalam ruang warna RGB.

4.2.2.2 Implementasi Kuantisasi Ruang Warna HSV

Setelah To dan Tb sudah dibentuk, proses selanjutnya adalah kuantisasi ruang warna HSV. Implementasi ini adalah implementasi dari *pseudocode* pada Gambar 3.4. Implementasi dalam matlab ditunjukkan oleh Kode Sumber 4.6.

1.	<code>h = inHSV(:, :, 1);</code>
2.	<code>h(h<=16) = h(h<=16)+360;</code>
3.	<code>h(h>342)= 0;</code>
4.	<code>h(h>16 & h <= 42)=1;</code>
5.	<code>h(h>42 & h <= 64)=2;</code>
6.	<code>h(h>64 & h <= 152)=3;</code>
7.	<code>h(h>152 & h <= 195)=4;</code>
8.	<code>h(h>195 & h <= 280)=5;</code>
9.	<code>h(h>280 & h <= 342)=6;</code>
10.	<code>s = inHSV(:, :, 2);</code>
11.	<code>s(s >= 0 & s < 0.3)=0;</code>
12.	<code>s(s >= 0.3 & s < 0.8)=1;</code>
13.	<code>s(s >= 0.8 & s < 1)=2;</code>
14.	<code>v = inHSV(:, :, 3);</code>
15.	<code>v(v >= 0 & v < 0.2)=0;</code>
16.	<code>v(v >= 0.2 & v < 0.7)=1;</code>
17.	<code>v(v >= 0.7 & v < 1)=2;</code>
18.	<code>outHSV = 9*h + 3*s + 1*v;</code>

Kode Sumber 4.6 Kode Implementasi Ruang Warna HSV

Pada baris 1-9 menunjukkan kuantisasi ruang warna HSV berdasarkan *hue channel* yang ditunjukkan Gambar 2.11 dan Persamaan (2. 5). Baris 10-17 adalah kuantisasi S dan V (*Saturation* dan *Value*) berdasarkan *SV channel* yang ditunjukkan pada Gambar 2.12 dan Persamaan (2. 6).

4.2.2.3 Implementasi Analisis Histogram

Analisis histogram bertujuan untuk menemukan warna dominan yang terdapat pada To dan Tb yang sudah dikuantisasi ruang warna HSV. Implementasi ini merupakan implementasi dari Gambar 3.5. Implementasi tersebut ditunjukkan pada Kode Sumber 4.7. Baris 1-6 menunjukkan perhitungan *global histogram* pada bagian yang sudah dikuantisasi ruang warna HSV dengan Persamaan (2. 9). Baris 10 menunjukkan proses mengidentifikasi setiap puncak pada *global histogram*.

1.	<code>k = zeros(1,63);</code>
2.	<code>for i=1:size(inVektor,1)</code>
3.	<code>for j=1:size(inVektor,2)</code>
4.	<code>if inMask(i,j) ==1</code>
5.	<code>k(inVektor(i,j)+1) = k(inVektor(i,j)+1) +1;</code>
6.	<code>end</code>
7.	<code>end</code>
8.	<code>end</code>
9.	
10.	<code>[~, peaks] = findpeaks(k);</code>

Kode Sumber 4.7 Kode Implementasi Analisis Histogram (Bagian 1)

12.	<code>maksP = max(k(peaks));</code>
13.	<code>minP = min(k(peaks));</code>
14.	<code>Rata2 = (maksP+minP)/2;</code>
15.	<code>StdDv = std(k(peaks));</code>
16.	<code>Tho = Rata2 - StdDv;</code>
17.	<code>Pho = [];</code>
18.	<code>for i = 1:size(peaks,2)</code>
19.	<code>if k(peaks(i)) > Tho</code>
20.	<code>Pho (end+1) = peaks(i);</code>

Kode Sumber 4.8 Kode Implementasi Analisis Histogram (Bagian 2)

Baris 13-20 pada Kode Sumber 4.8 menunjukkan implementasi menghapus puncak dari *global histogram* yang dihasilkan sebelumnya berdasarkan *threshold* (T_{ho}) yang dihitung dengan Persamaan (2. 10).

Baris 21-34 dari Kode Sumber 4.9 menunjukkan implementasi dari pemilihan puncak histogram berdasarkan *threshold* $T_{wo}=20$, ditentukan dengan asumsi bahwa tidak ada lagi warna dominan yang lebih dari pada 3 atau 4.

21.	<code>i = 1;</code>
22.	<code>if i==size(Pho,2)</code>
23.	<code>Pno(end+1) = Pho(i);</code>
24.	<code>End</code>
25.	<code>while i <=size(Pho,2)-1</code>
26.	<code>if Pho(i+1) - Pho(i) < Two</code>
27.	<code>if k(Pho(i+1)) > k(Pho(i))</code>
28.	<code>i = i+1;</code>
29.	<code>if i==size(Pho,2)</code>
30.	<code>Pno (end+1) =</code> <code>Pho(i);</code>
31.	<code>end</code>
32.	<code>else</code>
33.	<code>Pno (end+1) = Pho(i);</code>
34.	<code>i = i+2;</code>

**Kode Sumber 4.9 Kode Implementasi Analisis Histogram
(Bagian 3)**

4.2.3 Implementasi Ekstraksi Fitur

Pada bagian ini akan dijelaskan implementasi dari proses ekstraksi fitur. Implementasi ini bertujuan untuk menghasilkan nilai-nilai yang akan dijadikan vektor masukan pada *training* dan *testing* SVM. Fitur yang di ekstraksi adalah fitur warna, fitur *saliency*, fitur spasial dan fitur tekstur.

4.2.3.1 Fitur Warna

Implementasi fitur warna bertujuan untuk mengambil nilai warna r , g , dan b pada citra RGB dan *intensity* (i) pada citra yang sudah dikuantisasi HSV. Implementasi ini merupakan implementasi dari Gambar 3.8. Implementasi tersebut ditunjukkan pada Kode Sumber 4.10.

1.	<code>for i=1:size(TSpX,2)</code>
2.	<code> r(end+1) = img(TSpX(i),TSpY(i),1);</code>
3.	<code> g(end+1) = img(TSpX(i),TSpY(i),2);</code>
4.	<code> b (end+1) = img(TSpX(i),TSpY(i),3);</code>
5.	<code>End</code>

Kode Sumber 4.10 Kode Implementasi Ekstraksi Fitur Warna

Kode Sumber 4.10 digunakan untuk ekstraksi nilai warna pada citra RGB, untuk ekstraksi nilai *intensity* pada citra yang sudah dikuantisasi ruang warna HSV ditunjukkan Kode Sumber 4.11.

1.	<code>for i=1:size(TSpX,2)</code>
2.	<code> intensity (end+1) = LToHSV(TSpX(i),</code> <code> TSpY(i));</code>
3.	<code>end</code>

Kode Sumber 4.11 Kode Implementasi Ekstraksi Fitur Warna (*Intensity*)

4.2.3.2 Fitur *Saliency*

Implementasi fitur *saliency* bertujuan untuk mengambil nilai *saliency* pada citra *saliency map*, baik pada *training* piksel maupun *testing* piksel. Implementasi ini merupakan implementasi dari Gambar 3.9. Implementasi tersebut ditunjukkan pada Kode Sumber 4.12

1.	<code>s = [];</code>
2.	<code>for i=1 : size(TSpX,2)</code>
3.	<code>s(end+1) = imge(TSpX(i),TSpY(i));</code>

**Kode Sumber 4.12 Kode Implementasi Ekstraksi Fitur
*Saliency***

4.2.3.3 Fitur Spasial

Implementasi fitur spasial bertujuan untuk mengambil nilai posisi *x,y training* maupun *testing*. Implementasi ini merupakan implementasi dari Gambar 3.10. Implementasi tersebut ditunjukkan pada Kode Sumber 4.13.

1.	<code>[TSpX,TSpY]</code> <code>Homogeneity_Thresholding(XTo, YTo, LToHSV,</code> <code>Tlh);</code>	<code>=</code>
2.	<code>[TSnX, TSnY] = Homogeneity_Thresholding</code> <code>(XTb, YTb, LTbHSV, Tlh);</code>	

**Kode Sumber 4.13 Kode Implementasi Ekstraksi Fitur
Spasial pada *Training***

Untuk ekstraksi fitur spasial pada *testing* ditunjukkan pada Kode Sumber 4.14.

1.	<code>for i=1:size(RImg,1)</code>
2.	<code>for j=1:size(RImg,2)</code>
3.	<code>testx(end+1) = i;</code>
4.	<code>testy(end+1) =j;</code>

**Kode Sumber 4.14 Kode Implementasi Ekstraksi Fitur
Spasial pada *Testing***

4.2.3.4 Fitur Tekstur

Implementasi fitur tekstur bertujuan untuk mengambil nilai *local energy* dan *local gradient* pada citra yang sudah diaplikasikan *steerable filter*. Implementasi ini merupakan

implementasi dari Gambar 3.11. Implementasi tersebut ditunjukkan pada Kode Sumber 4.15. Di mana bari 1 menunjukkan implementasi konversi ruang warna RGB ke YCbCr dengan mengambil komponen *luminance* (Y). Baris 2-3 menunjukkan implementasi mengaplikasikan *steerable filter*. Baris 4 merupakan menghitung *local energy*, baris 5-8 menunjukkan implementasi dari perhitungan *local gradient*.

1.	<code>image = 0.2989*R + 0.5866*G + 0.114*B;</code>
2.	<code>for i = [1:length(theta)]</code>
3.	<code>temp = SteerableGaussian(I,theta(i),5,'false');</code>
4.	<code>localenergy = max([localenergy; reshape(abs(temp), 1, xImg * yImg)]);</code>
5.	<code>x = temp - temp_x(:, 2:end);</code>
6.	<code>y = temp - temp_y(2:end, :);</code>
7.	<code>g = sqrt(x.*x + y.*y);</code>
8.	<code>gradient = max([gradient; reshape(g, 1, xImg * yImg)]);</code>

Kode Sumber 4.15 Kode Implementasi Membentuk Nilai *Local Energy* dan *Local Gradient*

Steerable filter yang diimplementasikan menggunakan toolbox [9] ditunjukkan pada Kode Sumber 4.16

1.	<code>x = [-Wx:Wx];</code>
2.	<code>[xx,yy] = meshgrid(x,x);</code>
3.	<code>G1_0 = -(xx/sigma^2).*exp(-(xx.^2+yy.^2)/(2*sigma^2))/(sigma*sqrt(2*pi));</code>
4.	<code>G1_90 = -(yy/sigma^2).*exp(-(xx.^2+yy.^2)/(2*sigma^2))/(sigma*sqrt(2*pi));</code>
5.	<code>Ix = imfilter(I,G1_0,'same','replicate');</code>
6.	<code>Iy = imfilter(I,G1_90,'same','replicate');</code>
7.	<code>J = cos(theta)*Ix+sin(theta)*Iy;</code>

Kode Sumber 4.16 Kode Implementasi *Steerable Filter* [9]

4.2.4 Implementasi Klasifikasi Piksel dengan SVM

Pada bagian ini akan dijelaskan implementasi dari normalisasi data, *training* SVM, *testing* SVM, dan evaluasi dari hasil segmentasi.

4.2.4.1 Implementasi Normalisasi Data

Implementasi normalisasi data ini bertujuan untuk merubah nilai data dengan rentang nilai 0-1. Implementasi yang dilakukan berdasarkan Persamaan (2. 20). Implementasi ditunjukkan pada Kode Sumber 4.17.

1.	<code>datatrain = trainlist - min(trainlist(:));</code>
2.	<code>datatrain = (datatrain/range(datatrain(:)))*(maxtr-mintr);</code>
3.	<code>datatrain = datatrain - mintr;</code>

Kode Sumber 4.17 Kode Implementasi Normalisasi Data

4.2.4.2 Training SVM

Implementasi *training* SVM bertujuan untuk menghasilkan Model yang nantinya akan digunakan untuk *testing* SVM. Implementasi ini menggunakan *toolbox* [10] yang disediakan oleh matlab dengan Kode Sumber 4.18.

1.	<code>SVMModel = svmtrain(datatrain, trainclass, 'kernel_function', 'linear', 'boxconstrain', 0.1, 'autoscale', false);</code>
----	--

Kode Sumber 4.18 Kode Implementasi Pemanggilan Fungsi *svmtrain*

Fungsi *svmtrain* menghasilkan Model (*struct*) yang berisi *support vector*, nilai *alpha*, *bias*, fungsi kernel yang di gunakan,

index support vector, *group names*, dan *scale data*. Karena sebelumnya sudah dilakukan *scaling* data (normalisasi data) maka pada fungsi *svmtrain auto scale* diinisialisasi ‘*false*’ agar tidak melakukan *scaling* data lagi.

4.2.4.3 Testing SVM

Implementasi *testing* SVM bertujuan untuk mengklasifikasi data *testing* yang pada kasus ini adalah semua piksel pada citra. Implementasi ini menggunakan *toolbox* [11] yang disediakan oleh matlab. Ditunjukkan oleh Kode Sumber 4.19.

1.	SegmentasiCitra = svmclassify(SVMModel, datatest);
----	---

Kode Sumber 4.19 Kode Implementasi Pemanggilan Fungsi *svmclassify*

Hasil klasifikasi dari *testing* ini dikembalikan pada posisi citra awal, agar dapat membentuk citra biner. Implementasi dari mengembalikan hasil klasifikasi sesuai dengan posisi citra ditunjukkan Kode Sumber 4.20.

1.	hasil = zeros(size(Mo));
2.	for i=1:size(testx,2);
3.	if SegmentasiCitra(i) == 1
4.	hasil(testx(i), testy(i)) = 1;
5.	if SegmentasiCitra(i) == 0
6.	hasil(testx(i), testy(i)) = 0;
7.	End

Kode Sumber 4.20 Kode Implementasi Membentuk Citra Biner (Hasil Segmentasi)

4.2.4.4 Implementasi Evaluasi

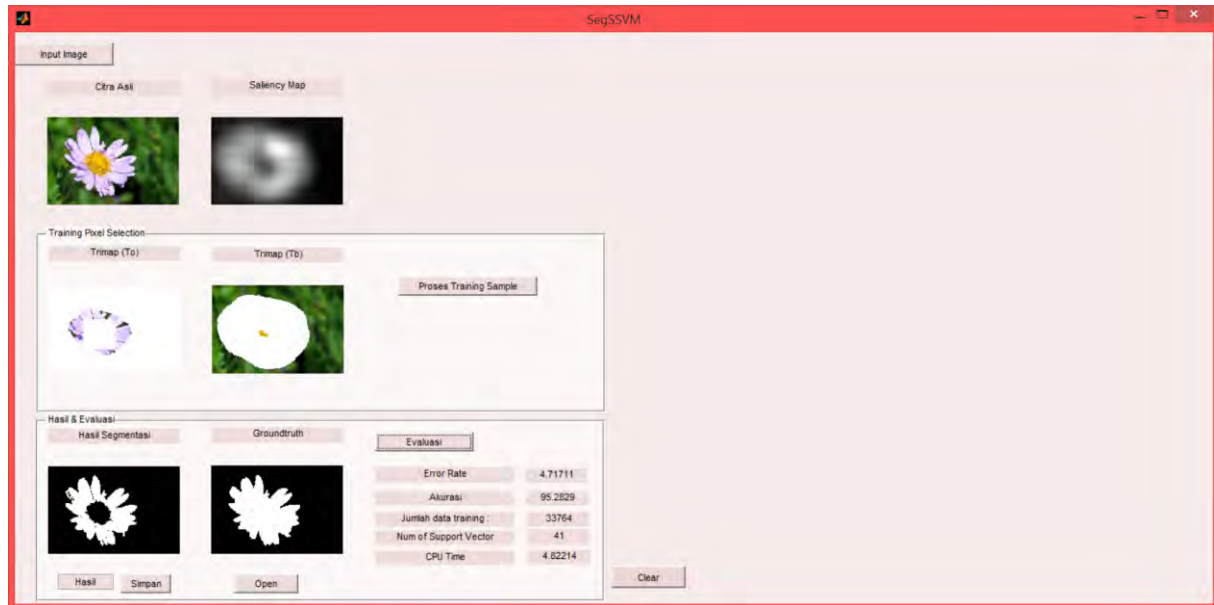
Implementasi *error rate* bertujuan untuk mencari kesalahan segmentasi pada hasil segmentasi jika dibandingkan dengan *ground truth*. Implementasi ini dilakukan berdasarkan Persamaan (2. 31) . Implementasinya ditunjukkan oleh Kode Sumber 4.21.

1.	<code>NfNm = seg == manual;</code>
2.	<code>NfNm = NfNm == 0;</code>
3.	<code>ER = sum(sum(NfNm)) / (size(NfNm, 1) * size(NfNm, 2)) * 100;</code>
4.	<code>akurasi = 100 - ER;</code>

Kode Sumber 4.21 Kode Implementasi Perhitungan Akurasi

4.3 Implementasi Antarmuka

Dalam implementasi antarmuka, aplikasi ini menampilkan satu jendela utama aplikasi. Semua proses yang terjadi mulai dari *visual saliency detection*, pembentukan *trimap*, klasifikasi piksel hingga evaluasi hasil segmentasi ditampilkan pada jendela utama yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Hasil Implementasi Antarmuka

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dilakukan uji coba sistem segmentasi citra menggunakan Saliency-SVM. Pembahasan meliputi lingkungan uji coba, skenario uji coba, dan hasil evaluasi. Uji coba yang dilakukan dengan memasukkan data masukan awal yaitu citra berwarna. Kemudian dilakukan proses segmentasi sesuai dengan masing-masing skenario sehingga dapat dihasilkan citra yang membedakan bagian objek dan latar belakang.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi metode *hybrid* Saliency-SVM untuk pemilihan data *training* secara otomatis pada segmentasi citra pada penelitian ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i5-4200U CPU @ 1.6 GHz
 - b. *Memory*(RAM): 4,00 GB
 - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: *Windows 8 single language 64-bit*
 - b. Perangkat pengembang: MATLAB 8.1.0 (R2013a).

5.2 Data Uji Coba

Data yang diuji pada Tugas Akhir ini adalah citra berwarna yang diambil dari [2] yang berisi 25.000 citra 20.000 *Image set A* dan 5000 *Image set B*. Pada tugas akhir ini diambil 50 citra secara acak yang ditunjukkan pada Tabel A. 1 dan Tabel A. 2.

5.3 Skenario dan Evaluasi Pengujian

Pada bagian ini dijelaskan mengenai skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu :

1. Pengujian awal dengan parameter yang sudah ditentukan.
2. Perbandingan hasil akurasi berdasarkan metode *resize* dan nilai n yang berbeda-beda pada matriks konvolusi *average filter* yang berbeda pada proses *visual saliency detection*.
3. Perbandingan hasil akurasi berdasarkan lebar piksel yang berbeda pada operasi morfologi dalam proses pra-segmentasi dan pembentukan *trimap*.
4. Perbandingan hasil akurasi berdasarkan nilai *threshold* pada *local homogeneity threshold*.
5. Perbandingan hasil akurasi berdasarkan fungsi kernel, parameter kernel dan nilai C pada proses klasifikasi piksel.

5.3.1 Uji coba dan Evaluasi Skenario 1

Skenario 1 adalah uji coba yang membandingkan hasil segmentasi dengan *ground truth* menggunakan nilai parameter dan metode berdasarkan literatur. Nilai parameter dan metode yang digunakan ditunjukkan oleh Tabel 5.1 dan Tabel 5.2.

Tabel 5.1 Parameter dan Metode Awal (Bagian 1)





No.	Parameter/ metode	Nilai/nama metode
1.	Metode <i>resize</i> pada <i>visual saliency detection</i>	' <i>bilinear</i> '
2.	Nilai n pada matriks konvolusi <i>average filter</i> pada <i>visual saliency detection</i>	3

Tabel 5.2 Parameter dan Metode Awal (Bagian 2)









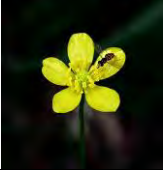
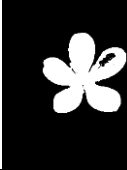


No.	Parameter/ metode	Nilai/ nama metode
3.	Lebar piksel pada operasi morfologi dalam proses pra-segmentasi dan pembentukan <i>trimap</i>	70
4.	<i>Threshold</i> pada <i>local homogeneity</i>	0
5.	<i>Local window</i> pada <i>local homogeneity</i>	3 x 3
6.	Fungsi kernel SVM pada proses klasifikasi piksel.	Linear
7.	Nilai parameter C SVM pada proses klasifikasi piksel.	1

Hasil dari uji coba ini ditunjukkan oleh Tabel 5.3 dan Tabel 5.4 dimana terdapat citra asli dan hasil segmentasi dari uji coba dengan menggunakan skenario 1.

Tabel 5.3 Hasil Uji Coba Skenario 1 (Bagian 1)

No.	Citra Asli	Hasil Segmentasi
1.		
2.		

Tabel 5.4 Hasil Uji Coba Skenario 1 (Bagian 2)

No.	Citra Asli		Hasil Segmentasi	
3.				
4.				
5.				
6.				
7.				
8.				





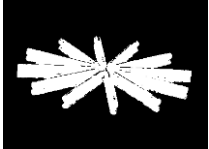
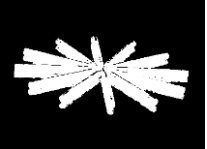



Hasil segmentasi dapat dilihat selengkapnya pada Tabel A. 3 dan Tabel A. 4 dan hasil citra pada tiap tahapnya ditunjukkan pada Tabel A. 5 sampai Tabel A. 29.

Hasil akurasi dengan skenario 1 dapat dilihat pada Tabel A. 30 Di mana memiliki rata-rata akurasi sebesar 93.69%, artinya segmentasi yang dihasilkan dengan parameter/ metode yang sudah ditentukan sudah cukup baik.





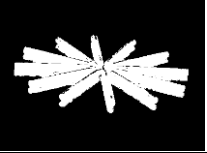
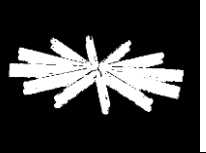



5.3.2 Uji Coba dan Evaluasi Skenario 2

Skenario 2 adalah uji coba yang dilakukan dengan membandingkan hasil segmentasi dengan *ground truth* berdasarkan perbedaan metode *resize* dan nilai n pada matriks konvolusi *average filter* yang berbeda pada proses *visual saliency detection*. Dengan kata lain uji coba skenario 2 ini dilakukan untuk mengevaluasi *visual saliency detection* jika dilakukan dengan nilai dan metode yang berbeda apakah mempengaruhi hasil segmentasi atau tidak.





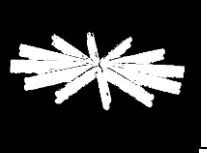
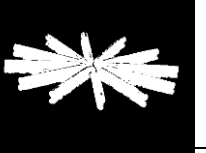

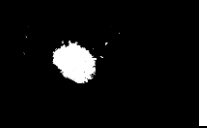

Tabel 5.5 Hasil Uji Coba Skenario 2 (Bagian 1)

No.	Citra Asli	<i>bilinear</i> $n = 3$	<i>bicubic</i> $n = 3$
1.			
2.			
3.			

Tabel 5.6 Hasil Uji Coba Skenario 2 (Bagian 2)

No.	Citra Asli	<i>Bilinear n = 5</i>	<i>Bicubic n = 5</i>
1.			
2.			
3.			

Tabel 5.7 Hasil Uji Coba Skenario 2 (Bagian 3)

No.	Citra Asli	<i>Bilinear n = 7</i>	<i>Bicubic n = 7</i>
1.			
2.			
3.			

Tabel 5.8 Hasil Rata-Rata Akurasi Uji Coba Skenario 2

Metode <i>resize</i>	Konvolusi Matriks <i>Average filter</i>		
	3	5	7
<i>Bilinear</i>	93.69	93.61	93.54
<i>Bicubic</i>	93.65	93.53	93.52

Perbandingan metode *resize* yang digunakan pada *visual saliency detection* adalah *bilinear* dan *bicubic*. Sedangkan nilai n pada matriks konvolusi *average filter* adalah 3 di mana terdapat pada literatur acuan, 5 dan 7. Hasil segmentasi dari uji coba 2 ditunjukkan oleh Tabel 5.5, Tabel 5.6, dan Tabel 5.7. Serta hasil rata-rata akurasi ditunjukkan Tabel 5.8.


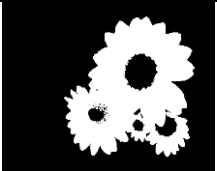
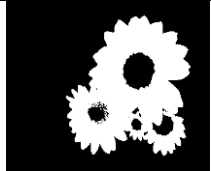

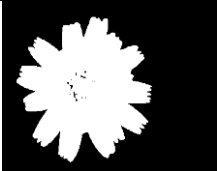
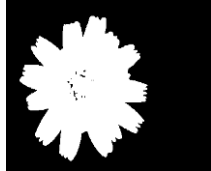

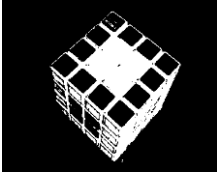
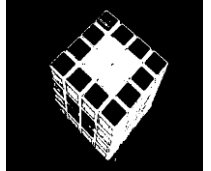






Dari hasil akurasi uji coba skenario 2 ini dapat dilihat pada Tabel A. 31, Tabel A. 32, Tabel A. 33, dan Tabel A. 34 bahwa perbedaan hasil segmentasi antara metode *resize bilinear* dan *bicubic* tidak begitu signifikan dan begitu juga dengan nilai n pada konvolusi matriks *average filter*. Tetapi hasil terbaik dihasilkan oleh metode *resize bilinear* dengan nilai n konvolusi matriks *average filter* adalah 3 dengan rata-rata akurasi 93.69%. Parameter dan metode yang menghasilkan rata-rata akurasi pada uji coba skenario 2 digunakan pada uji coba skenario selanjutnya.

5.3.3 Uji Coba dan Evaluasi Skenario 3

Skenario uji coba 3 adalah uji coba yang dilakukan dengan membandingkan hasil segmentasi dengan *ground truth* berdasarkan perbedaan lebar piksel pada operasi morfologi pada proses pra-segmentasi dan pembentukan *trimap* yang berbeda. Di mana percobaan ini dilakukan pada lebar piksel pada operasi morfologi dengan nilai 70 dan 75. Hasil segmentasi dengan lebar piksel yang berbeda ditunjukkan pada Tabel 5.9. Rata-rata akurasi dengan lebar piksel 70 adalah 93.69% dan lebar piksel 75 adalah 93.82%. Dari hasil akurasi yang ditunjukkan pada Tabel A. 36 dan Tabel A. 37, hasil akurasi terbaik dihasilkan oleh lebar piksel 75,

maka untuk skenario uji coba selanjutnya digunakan lebar piksel terbaik tersebut.

















Tabel 5.9 Hasil Uji Coba Skenario 3

No .	Citra Asli	70	75
1.			
2.			
3.			
4.			
5.			

5.3.4 Uji Coba dan Evaluasi Skenario 4

Skenario 4 adalah uji coba yang dilakukan dengan membandingkan hasil segmentasi dengan *ground truth* berdasarkan perbedaan nilai *threshold* pada proses *local homogeneity threshold*, di mana nilai *threshold* ini dapat mempengaruhi jumlah data *training* yang dihasilkan dalam klasifikasi piksel untuk menghasilkan segmentasi citra yang ditunjukkan pada Tabel A. 38. Nilai *threshold* yang digunakan pada uji coba ini adalah 0, 50 dan 100. Hasil segmentasi dengan nilai *threshold* yang berbeda-beda ditunjukkan oleh Tabel 5.10 dan rata-rata akurasi ditunjukkan oleh Tabel 5.11 untuk hasil akurasi selengkapnya ditunjukkan pada Tabel A. 39, Tabel A. 40, dan Tabel A. 41. Di mana akurasi terbaik dihasilkan dengan nilai *threshold* 100 yaitu 94.26%. Maka untuk uji coba skenario selanjutnya digunakan nilai *threshold* 100.

Tabel 5.10 Hasil Uji Coba Skenario 4

No .	Citra Asli	0	50	100
1.				
2.				
3.				
4.				











Tabel 5.11 Hasil Rata-Rata Akurasi Uji Coba Skenario 3

Nilai <i>threshold</i>	0	50	100
Akurasi	93.82	94.23	94.26











5.3.5 Uji Coba dan Evaluasi Skenario 5

Skenario 5 adalah uji coba yang dilakukan dengan membandingkan hasil segmentasi dengan *ground truth* berdasarkan perbedaan fungsi kernel dan nilai parameter C pada klasifikasi piksel untuk menghasilkan segmentasi citra. Fungsi kernel yang digunakan pada uji coba ini adalah linear, polinomial dan rbf dengan parameter C dengan nilai 0.1 dan 1.

Tabel 5.12 Hasil Segmentasi Uji Coba Skenario 5 (Bagian 1)

Nilai C	Linear	Polinomial (3)	Polinomial (4)	Rbf(0.25)	Rbf(0.5)
0.1					
1					











Tabel 5.13 Hasil Segmentasi Uji Coba Skenario 5 (Bagian 2)

Nilai C	Linear	Polinomial (3)	Polinomial (4)	Rbf(0.25)	Rbf(0.5)
0.1					
1					

Tabel 5.14 Hasil Segmentasi Uji Coba Skenario 5 (Bagian 3)

Hasil segmentasi citra ditunjukkan oleh Tabel 5.12, Tabel 5.13 dan Tabel 5.14. Hasil rata-rata akurasi dari uji coba skenario

5 ditunjukkan oleh Tabel 5.15 dan hasil akurasi selengkapnya ditunjukkan pada Tabel A. 42 sampai Tabel A. 51. Di mana hasil akurasi terbaik dihasilkan oleh fungsi kernel rbf dengan nilai C adalah 1 dengan rata-rata akurasi 94.84%.

Nilai C	Linear	Polinomial (3)	Polinomial (4)	Rbf(0.25)	Rbf(0.5)
0.1					
1					

Tabel 5.15 Hasil Rata-Rata Akurasi Uji Coba Skenario 5

Fungsi Kernel	Parameter	Nilai C	
		0.1	1
Linear		94.45	94.26
Polinomial	$p = 3$	94.10	94.23
	$p = 4$	94.15	94.20
Rbf	$\sigma = 0.25$	93.45	93.35
	$\sigma = 0.5$	94.71	94.84

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Metode Saliency-SVM dapat menghasilkan data *training* dan melakukan segmentasi citra berwarna dengan baik dengan rata-rata akurasi mencapai 94.84%.
2. Hasil segmentasi citra dengan Saliency-SVM dipengaruhi oleh metode *resize* dan nilai n pada matriks konvolusi *average filter* dalam proses *visual saliency detection* walaupun dengan perbedaan akurasi yang tidak signifikan. Akurasi terbaik dihasilkan metode *resize bilinear* dan nilai n adalah 3.
3. Hasil segmentasi citra dengan Saliency-SVM dipengaruhi oleh lebar piksel pada operasi morfologi dalam proses pra-segmentasi dan pembentukan *trimap*, yaitu memiliki akurasi terbaik dengan lebar piksel adalah 75.
4. Hasil segmentasi citra dengan Saliency-SVM dipengaruhi oleh nilai *threshold* pada pemilihan *training* piksel dengan *local homogeneity threshold*, di mana jumlah data *training* yang dihasilkan mempengaruhi hasil segmentasi citra. Akurasi terbaik dihasilkan oleh nilai *threshold* 100, yang artinya semakin banyak *training* piksel yang terbentuk semakin baik segmentasi citra yang dihasilkan.
5. Hasil segmentasi citra dengan Saliency-SVM juga dipengaruhi oleh fungsi kernel dan parameter C pada proses klasifikasi

piksel dengan SVM. Akurasi terbaik dihasilkan oleh fungsi kernel Rbf dengan nilai σ 0.5 dan nilai parameter C adalah 1.

6.2. Saran

Beberapa saran yang hendak disampaikan terkait dengan pengerjaan tugas akhir ini adalah:

1. Aplikasi ini menghasilkan akurasi yang tidak maksimal pada citra dengan objek dan latar belakang dengan warna yang rumit atau tidak spesifik maka dibutuhkan metode yang dapat menangani hal tersebut.
2. Diperlukan penentuan parameter yang baik dan kriteria yang lebih banyak pada proses pemilihan data *training* untuk mencegah terjadinya *overlap* pada pemilihan data *training*.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Metode Saliency-SVM dapat menghasilkan data *training* dan melakukan segmentasi citra berwarna dengan baik dengan rata-rata akurasi mencapai 94.84%.
2. Hasil segmentasi citra dengan Saliency-SVM dipengaruhi oleh metode *resize* dan nilai n pada matriks konvolusi *average filter* dalam proses *visual saliency detection* walaupun dengan perbedaan akurasi yang tidak signifikan. Akurasi terbaik dihasilkan metode *resize bilinear* dan nilai n adalah 3.
3. Hasil segmentasi citra dengan Saliency-SVM dipengaruhi oleh lebar piksel pada operasi morfologi dalam proses pra-segmentasi dan pembentukan *trimap*, yaitu memiliki akurasi terbaik dengan lebar piksel adalah 75.
4. Hasil segmentasi citra dengan Saliency-SVM dipengaruhi oleh nilai *threshold* pada pemilihan *training* piksel dengan *local homogeneity threshold*, di mana jumlah data *training* yang dihasilkan mempengaruhi hasil segmentasi citra. Akurasi terbaik dihasilkan oleh nilai *threshold* 100, yang artinya semakin banyak *training* piksel yang terbentuk semakin baik segmentasi citra yang dihasilkan.
5. Hasil segmentasi citra dengan Saliency-SVM juga dipengaruhi oleh fungsi kernel dan parameter C pada proses klasifikasi

piksel dengan SVM. Akurasi terbaik dihasilkan oleh fungsi kernel Rbf dengan nilai σ 0.5 dan nilai parameter C adalah 1.































6.2. Saran

Beberapa saran yang hendak disampaikan terkait dengan pengerjaan tugas akhir ini adalah:
















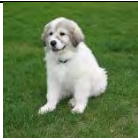




1. Aplikasi ini menghasilkan akurasi yang tidak maksimal pada citra dengan objek dan latar belakang dengan warna yang rumit atau tidak spesifik maka dibutuhkan metode yang dapat menangani hal tersebut.
2. Diperlukan penentuan parameter yang baik dan kriteria yang lebih banyak pada proses pemilihan data *training* untuk mencegah terjadinya *overlap* pada pemilihan data *training*.

LAMPIRAN
















Tabel A. 1 Citra Masukan (Bagian 1)

 Test1.jpg	 Test2.jpg	 Test3.jpg	 Test4.jpg	 Test5.jpg
 Test6.jpg	 Test7.jpg	 Test9.jpg	 Test10.jpg	 Test11.jpg
 Test12.jpg	 Test13.jpg	 Test15.jpg	 Test16.jpg	 Test17.jpg
 Test18.jpg	 Test19.jpg	 Test20.jpg	 Test21.jpg	 Test22.jpg
 Test23.jpg	 Test25.jpg	 Test26.jpg	 Test27.jpg	 Test29.jpg
 Test31.jpg	 Test32.jpg	 Test33.jpg	 Test36.jpg	 Test37.jpg

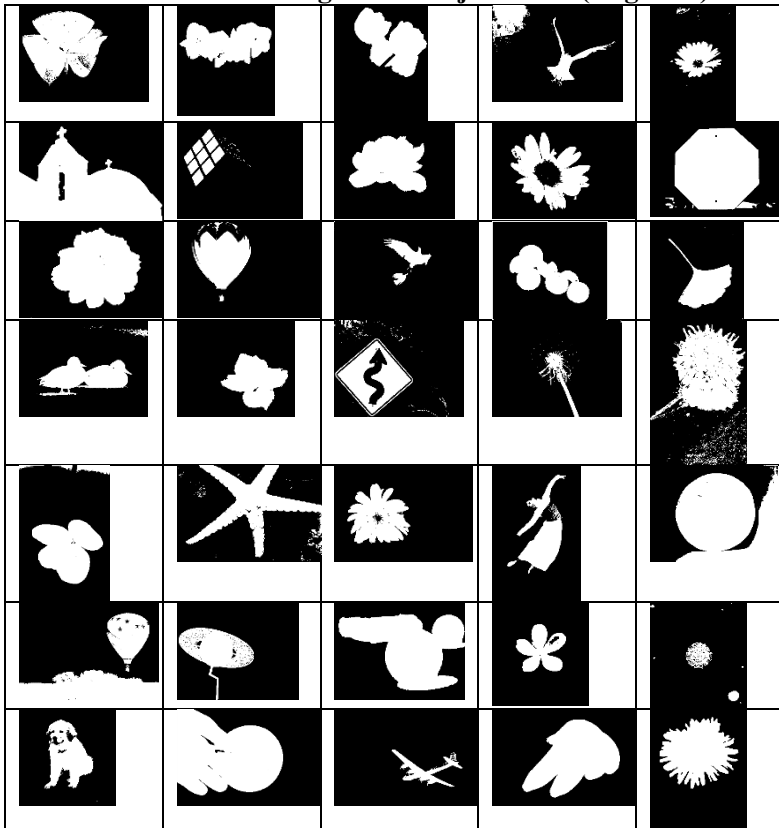
Tabel A. 2 Citra Masukan (Bagian 2)

 Test38.jpg	 Test40.jpg	 Test41.jpg	 Test42.jpg	 Test43.jpg
 Test44.jpg	 Test45.jpg	 Test46.jpg	 Test47.jpg	 Test48.jpg
 Test49.jpg	 Test50.jpg	 Test51.jpg	 Test52.jpg	 Test53.jpg
 Test54.jpg	 Test55.jpg	 Test56.jpg	 Test57.jpg	 Test58.jpg


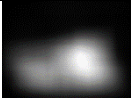






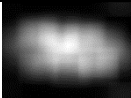



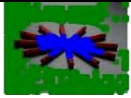

Tabel A. 3 Hasil Segmentasi Uji Coba 1 (Bagian 1)


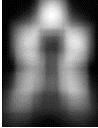










Tabel A. 4 Hasil Segmentasi Uji Coba 1 (Bagian 2)




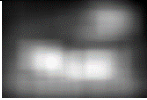






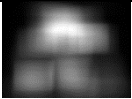





Tabel A. 5 Hasil Citra dari Tahapan-Tahapan (Bagian 1)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


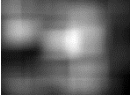










Tabel A. 6 Hasil Citra dari Tahapan-Tahapan (Bagian 2)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i> <i>Tb</i>	 Titik biru adalah lokasi <i>TSp</i> dan Titik hijau adalah <i>TSn</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i> <i>Tb</i>		


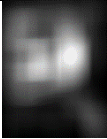



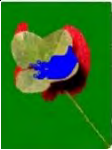


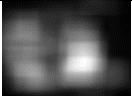





Tabel A. 7 Hasil Citra dari Tahapan-Tahapan (Bagian 3)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


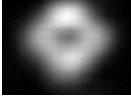












Tabel A. 8 Hasil Citra dari Tahapan-Tahapan (Bagian 4)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i> <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i> <i>Tb</i>		









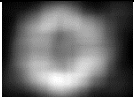



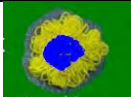

Tabel A. 9 Hasil Citra dari Tahapan-Tahapan (Bagian 5)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM
			 <i>To</i>  <i>Tb</i>		


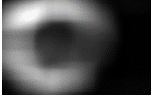






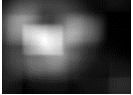





Tabel A. 10 Hasil Citra dari Tahapan-Tahapan (Bagian 6)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


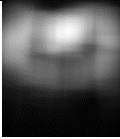






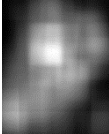





Tabel A. 11 Hasil Citra dari Tahapan-Tahapan (Bagian 7)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


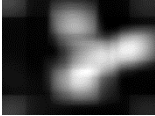





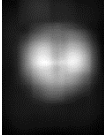




Tabel A. 12 Hasil Citra dari Tahapan-Tahapan (bagian 8)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


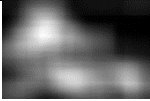





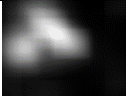

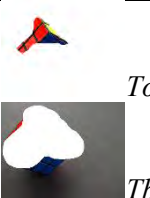


Tabel A. 13 Hasil Citra dari Tahapan-Tahapan (Bagian 9)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TSp</i> dan Titik hijau adalah <i>TSn</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


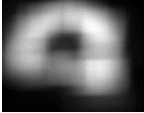






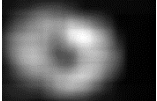





Tabel A. 14 Hasil Citra dari Tahapan-Tahapan (Bagian 10)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			<p><i>To</i></p>  <p><i>Tb</i></p>	 <p>Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i>.</p>	 <p>Hasil klasifikasi piksel dengan SVM.</p>
			<p><i>To</i></p>  <p><i>Tb</i></p>		









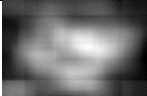





Tabel A. 15 Hasil Citra dari Tahapan-Tahapan (Bagian 11)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <p><i>To</i> <i>Tb</i></p>	 <p>Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i>.</p>	 <p>Hasil klasifikasi piksel dengan SVM.</p>
			 <p><i>To</i> <i>Tb</i></p>		


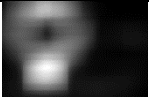

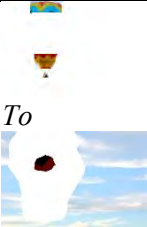



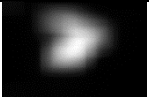




Tabel A. 16 Hasil Citra dari Tahapan-Tahapan (Bagian 13)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


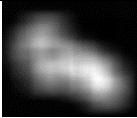






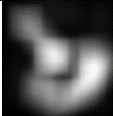





Tabel A. 17 Hasil Citra dari Tahapan-Tahapan (Bagian 14)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


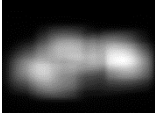








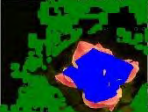

Tabel A. 18 Hasil Citra dari Tahapan-Tahapan (Bagian 15)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i> <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i> <i>Tb</i>		


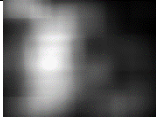






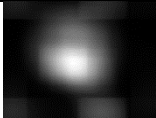





Tabel A. 19 Hasil Citra dari Tahapan-Tahapan (Bagian 16)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 		


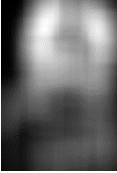





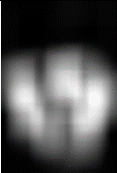

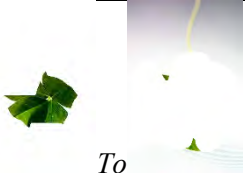


Tabel A. 20 Hasil Citra dari Tahapan-Tahapan (Bagian 17)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			<p><i>To</i></p>  <p><i>Tb</i></p>	 <p>Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i>.</p>	 <p>Hasil klasifikasi piksel dengan SVM.</p>
			<p><i>To</i></p>  <p><i>Tb</i></p>		


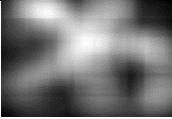






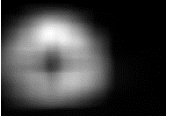


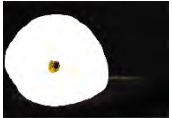


Tabel A. 21 Hasil Citra dari Tahapan-Tahapan (Bagian 18)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


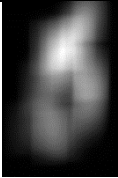






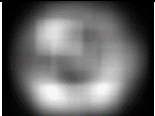


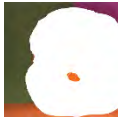


Tabel A. 22 Hasil Citra dari Tahapan-Tahapan (Bagian 19)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training piksel</i>	Hasil Segmentasi
				 Titik biru adalah lokasi <i>TSp</i> dan Titik hijau adalah <i>TSn</i> .	 Hasil klasifikasi piksel dengan SVM.
					


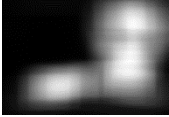

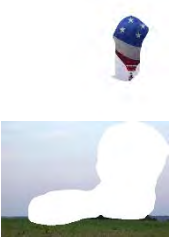





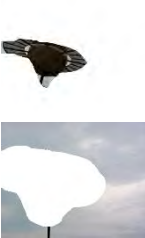
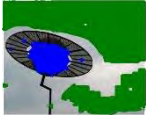
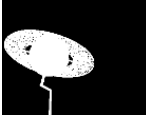
Tabel A. 23 Hasil Citra dari Tahapan-Tahapan (Bagian 20)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


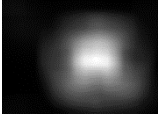


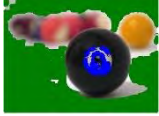


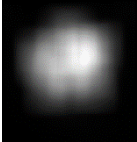




Tabel A. 24 Hasil Citra dari Tahapan-Tahapan (Bagian 21)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TSp</i> dan Titik hijau adalah <i>TSn</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i>  <i>Tb</i>		


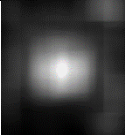
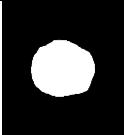





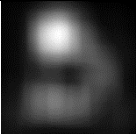





Tabel A. 25 Hasil Citra dari Tahapan-Tahapan (Bagian 21)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i> <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i> <i>Tb</i>		


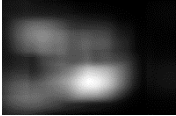





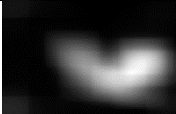




Tabel A. 26 Hasil Citra dari Tahapan-Tahapan (Bagian 22)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i> <i>Tb</i>	 Titik biru adalah lokasi <i>TSp</i> dan Titik hijau adalah <i>TSn</i> .	 Hasil klasifikasi piksel dengan SVM.
			 <i>To</i> <i>Tb</i>		


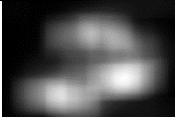





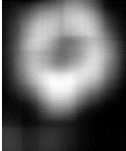

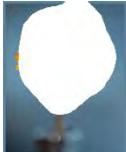


Tabel A. 27 Hasil Citra dari Tahapan-Tahapan (Bagian 23)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			 <i>To</i>  <i>Tb</i>	 Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i> .	
			 <i>To</i>  <i>Tb</i>		

Tabel A. 28 Hasil Citra dari Tahapan-Tahapan (Bagian 24)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			<p><i>To</i></p>  <p><i>Tb</i></p>	 <p>Titik biru adalah lokasi <i>TS_p</i> dan Titik hijau adalah <i>TS_n</i>.</p>	 <p>Hasil klasifikasi piksel dengan SVM.</p>
			<p><i>To</i></p>  <p><i>Tb</i></p>		

Tabel A. 29 Hasil Citra dari Tahapan-Tahapan (Bagian 25)

Citra Asli	<i>Saliency Map</i>	Citra Biner	<i>Trimap</i>	<i>Training</i> piksel	Hasil Segmentasi
			<p><i>To</i></p>  <p><i>Tb</i></p>	 <p>Titik biru adalah lokasi <i>TSp</i> dan Titik hijau adalah <i>TSn</i>.</p>	 <p>Hasil klasifikasi piksel dengan SVM.</p>
			<p><i>To</i></p>  <p><i>Tb</i></p>		

Tabel A. 30 Akurasi Uji Coba Skenario 1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.06	Test 20	99.56	Test 43	92.44
Test 2	98.70	Test 21	92.73	Test 44	97.36
Test 3	94.28	Test 22	98.68	Test 45	95.68
Test 4	96.87	Test 23	99.24	Test 46	98.75
Test 5	94.69	Test 25	84.55	Test 47	95.22
Test 6	87.98	Test 26	99.53	Test 48	78.75
Test7	91.44	Test 27	95.28	Test 49	76.17
Test 9	95.51	Test 29	98.49	Test 50	99.17
Test 10	97.29	Test 31	99.22	Test 51	72.87
Test 11	98.75	Test 32	95.19	Test 52	99.21
Test 12	85.89	Test 33	98.48	Test 53	77.88
Test 13	95.71	Test 36	97.18	Test 54	96.66
Test 15	82.67	Test 37	99.64	Test 55	74.85
Test 16	97.77	Test 38	97.96	Test 56	98.51
Test 17	97.37	Test 40	99.55	Test 57	99.31
Test 18	98.06	Test 41	91.77	Test 58	97.51
Test 19	99.28	Test 42	85.56	Rata-rata	93.69

Tabel A. 31 Hasil Akurasi Uji Coba Skenario 2 (Bagian 1) – *bilinear* $n=5$

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.61	Test 20	99.56	Test 43	92.49
Test 2	98.70	Test 21	99.54	Test 44	97.36
Test 3	94.39	Test 22	98.68	Test 45	95.70
Test 4	96.87	Test 23	99.24	Test 46	98.76
Test 5	94.79	Test 25	84.46	Test 47	95.23
Test 6	92.09	Test 26	99.52	Test 48	78.48
Test7	92.68	Test 27	79.06	Test 49	76.18
Test 9	95.24	Test 29	98.55	Test 50	99.17
Test 10	97.28	Test 31	99.23	Test 51	72.87
Test 11	98.75	Test 32	95.19	Test 52	99.21
Test 12	85.85	Test 33	98.48	Test 53	77.88
Test 13	95.72	Test 36	97.18	Test 54	96.66
Test 15	82.39	Test 37	99.65	Test 55	74.85
Test 16	97.77	Test 38	98.22	Test 56	98.51
Test 17	97.34	Test 40	99.55	Test 57	99.33
Test 18	98.02	Test 41	91.79	Test 58	97.53
Test 19	99.28	Test 42	85.56	Rata-Rata	93.61

Tabel A. 32 Hasil Akurasi Uji Coba Skenario 2 (Bagian 2) – *bilinear* $n=7$

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.61	Test 20	99.56	Test 43	92.49
Test 2	98.66	Test 21	99.54	Test 44	97.37
Test 3	94.50	Test 22	98.68	Test 45	95.70
Test 4	96.87	Test 23	99.24	Test 46	98.76
Test 5	94.78	Test 25	84.54	Test 47	95.23
Test 6	92.09	Test 26	99.53	Test 48	77.88
Test7	91.38	Test 27	79.06	Test 49	76.18
Test 9	95.23	Test 29	98.50	Test 50	99.17
Test 10	97.28	Test 31	99.23	Test 51	70.58
Test 11	98.75	Test 32	95.19	Test 52	99.21
Test 12	85.85	Test 33	98.48	Test 53	77.88
Test 13	95.72	Test 36	97.18	Test 54	96.66
Test 15	83.03	Test 37	99.64	Test 55	74.85
Test 16	97.78	Test 38	98.22	Test 56	98.51
Test 17	97.34	Test 40	99.55	Test 57	99.49
Test 18	98.04	Test 41	91.79	Test 58	97.52
Test 19	99.28	Test 42	85.48	Rata-Rata	93.54

Tabel A. 33 Hasil Akurasi Uji Coba Skenario 2 (Bagian 3) – *bicubic* $n = 3$

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.04	Test 20	99.56	Test 43	92.49
Test 2	98.69	Test 21	92.73	Test 44	97.35
Test 3	94.26	Test 22	98.68	Test 45	95.70
Test 4	96.87	Test 23	99.24	Test 46	98.75
Test 5	93.88	Test 25	84.55	Test 47	95.22
Test 6	87.96	Test 26	99.53	Test 48	77.81
Test7	91.44	Test 27	95.31	Test 49	76.18
Test 9	95.51	Test 29	98.56	Test 50	99.17
Test 10	97.29	Test 31	99.24	Test 51	72.88
Test 11	98.75	Test 32	95.19	Test 52	99.19
Test 12	85.89	Test 33	98.48	Test 53	77.88
Test 13	95.86	Test 36	97.19	Test 54	96.66
Test 15	82.66	Test 37	99.64	Test 55	74.85
Test 16	97.77	Test 38	97.96	Test 56	98.51
Test 17	97.37	Test 40	99.55	Test 57	99.31
Test 18	98.06	Test 41	91.77	Test 58	97.52
Test 19	99.18	Test 42	85.56	Rata-Rata	93.65

Tabel A. 34 Hasil Akurasi Uji Coba Skenario 2 (Bagian 4) –*bicubic* $n=5$

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.60	Test 20	99.56	Test 43	92.49
Test 2	98.70	Test 21	99.54	Test 44	97.36
Test 3	94.75	Test 22	98.67	Test 45	95.85
Test 4	96.87	Test 23	99.23	Test 46	98.75
Test 5	93.43	Test 25	84.46	Test 47	95.22
Test 6	88.00	Test 26	99.52	Test 48	78.35
Test7	92.81	Test 27	79.06	Test 49	76.18
Test 9	95.51	Test 29	98.61	Test 50	99.17
Test 10	97.29	Test 31	99.25	Test 51	72.87
Test 11	98.75	Test 32	95.19	Test 52	99.21
Test 12	85.85	Test 33	98.48	Test 53	77.88
Test 13	95.71	Test 36	97.18	Test 54	96.66
Test 15	82.97	Test 37	99.65	Test 55	74.85
Test 16	97.77	Test 38	98.21	Test 56	98.51
Test 17	97.37	Test 40	99.55	Test 57	99.33
Test 18	98.04	Test 41	91.77	Test 58	97.51
Test 19	99.17	Test 42	85.56	Rata-Rata	93.53

Tabel A. 35 Hasil Akurasi Uji Coba Skenario 2 (Bagian 5) –*bicubic* $n=7$

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.60	Test 20	99.55	Test 43	92.55
Test 2	98.70	Test 21	99.54	Test 44	97.37
Test 3	94.42	Test 22	98.68	Test 45	95.85
Test 4	96.88	Test 23	99.23	Test 46	98.76
Test 5	93.76	Test 25	84.54	Test 47	95.23
Test 6	92.09	Test 26	99.53	Test 48	77.79
Test7	91.36	Test 27	79.06	Test 49	76.18
Test 9	95.21	Test 29	98.50	Test 50	99.17
Test 10	97.28	Test 31	99.25	Test 51	70.59
Test 11	98.76	Test 32	95.19	Test 52	99.21
Test 12	85.85	Test 33	98.48	Test 53	77.88
Test 13	95.74	Test 36	97.19	Test 54	96.66
Test 15	83.02	Test 37	99.65	Test 55	74.85
Test 16	97.78	Test 38	98.21	Test 56	98.51
Test 17	97.34	Test 40	99.55	Test 57	99.50
Test 18	98.04	Test 41	91.79	Test 58	97.51
Test 19	99.17	Test 42	85.48	Rata-Rata	93.52














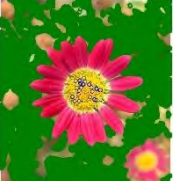


Tabel A. 36 Hasil Akurasi Uji Coba Skenario 3 (Bagian 1) – lebar piksel = 70

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.06	Test 20	99.56	Test 43	92.44
Test 2	98.70	Test 21	92.73	Test 44	97.36
Test 3	94.28	Test 22	98.68	Test 45	95.68
Test 4	96.87	Test 23	99.24	Test 46	98.75
Test 5	94.69	Test 25	84.55	Test 47	95.22
Test 6	87.98	Test 26	99.53	Test 48	78.75
Test7	91.44	Test 27	95.28	Test 49	76.17
Test 9	95.51	Test 29	98.49	Test 50	99.17
Test 10	97.29	Test 31	99.22	Test 51	72.87
Test 11	98.75	Test 32	95.19	Test 52	99.21
Test 12	85.89	Test 33	98.48	Test 53	77.88
Test 13	95.71	Test 36	97.18	Test 54	96.66
Test 15	82.67	Test 37	99.64	Test 55	74.85
Test 16	97.77	Test 38	97.96	Test 56	98.51
Test 17	97.37	Test 40	99.55	Test 57	99.31
Test 18	98.06	Test 41	91.77	Test 58	97.51
Test 19	99.28	Test 42	85.56	Rata-rata	93.69

Tabel A. 37 Hasil Akurasi Uji Coba Skenario 3 (Bagian 2) –lebar piksel = 75

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	88.42	Test 20	99.56	Test 43	92.59
Test 2	98.44	Test 21	99.53	Test 44	97.40
Test 3	91.75	Test 22	98.68	Test 45	95.61
Test 4	96.87	Test 23	99.24	Test 46	98.74
Test 5	94.79	Test 25	84.53	Test 47	95.22
Test 6	87.91	Test 26	99.55	Test 48	80.40
Test7	91.45	Test 27	95.31	Test 49	76.18
Test 9	95.16	Test 29	98.68	Test 50	99.17
Test 10	97.29	Test 31	99.23	Test 51	72.88
Test 11	98.75	Test 32	96.79	Test 52	99.21
Test 12	85.89	Test 33	98.48	Test 53	77.89
Test 13	96.14	Test 36	97.19	Test 54	96.66
Test 15	82.52	Test 37	99.64	Test 55	74.84
Test 16	97.77	Test 38	97.76	Test 56	98.51
Test 17	97.37	Test 40	99.55	Test 57	99.39
Test 18	98.03	Test 41	91.74	Test 58	97.54
Test 19	99.26	Test 42	85.48	Rata-rata	93.82

Tabel A. 38 Data *Training* Sesuai Nilai *Threshold*

Citra	0	50	100
			
			
			
			

Tabel A. 39 Hasil Akurasi Uji Coba Skenario 4 (Bagian 1) – *Threshold* = 0

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	88.42	Test 20	99.56	Test 43	92.59
Test 2	98.44	Test 21	99.53	Test 44	97.40
Test 3	91.75	Test 22	98.68	Test 45	95.61
Test 4	96.87	Test 23	99.24	Test 46	98.74
Test 5	94.79	Test 25	84.53	Test 47	95.22
Test 6	87.91	Test 26	99.55	Test 48	80.40
Test7	91.45	Test 27	95.31	Test 49	76.18
Test 9	95.16	Test 29	98.68	Test 50	99.17
Test 10	97.29	Test 31	99.23	Test 51	72.88
Test 11	98.75	Test 32	96.79	Test 52	99.21
Test 12	85.89	Test 33	98.48	Test 53	77.89
Test 13	96.14	Test 36	97.19	Test 54	96.66
Test 15	82.52	Test 37	99.64	Test 55	74.84
Test 16	97.77	Test 38	97.76	Test 56	98.51
Test 17	97.37	Test 40	99.55	Test 57	99.39
Test 18	98.03	Test 41	91.74	Test 58	97.54
Test 19	99.26	Test 42	85.48	Rata-rata	93.82

Tabel A. 40 Hasil Akurasi Uji Coba Skenario 4 (Bagian 2) – *Threshold* = 50

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	87.57	Test 20	99.55	Test 43	93.70
Test 2	98.71	Test 21	99.54	Test 44	94.48
Test 3	91.72	Test 22	98.61	Test 45	95.92
Test 4	96.83	Test 23	99.23	Test 46	99.36
Test 5	97.89	Test 25	84.61	Test 47	95.39
Test 6	95.52	Test 26	99.56	Test 48	81.18
Test7	91.75	Test 27	95.48	Test 49	76.20
Test 9	95.71	Test 29	98.38	Test 50	99.20
Test 10	97.54	Test 31	99.28	Test 51	72.37
Test 11	98.85	Test 32	96.76	Test 52	99.23
Test 12	85.51	Test 33	98.49	Test 53	92.27
Test 13	93.57	Test 36	97.24	Test 54	97.16
Test 15	82.48	Test 37	99.66	Test 55	74.72
Test 16	97.65	Test 38	97.27	Test 56	98.48
Test 17	97.18	Test 40	99.56	Test 57	99.24
Test 18	98.32	Test 41	90.04	Test 58	97.71
Test 19	99.21	Test 42	85.64	Rata-rata	94.23

Tabel A. 41 Hasil Akurasi Uji Coba Skenario 4 (Bagian 3) – *Threshold* = 100

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	87.31	Test 20	99.54	Test 43	93.79
Test 2	98.70	Test 21	99.54	Test 44	95.85
Test 3	91.71	Test 22	98.61	Test 45	95.93
Test 4	96.82	Test 23	99.23	Test 46	99.35
Test 5	97.89	Test 25	84.62	Test 47	95.39
Test 6	95.93	Test 26	99.56	Test 48	81.09
Test7	91.75	Test 27	95.49	Test 49	76.20
Test 9	95.73	Test 29	98.38	Test 50	99.20
Test 10	97.54	Test 31	99.30	Test 51	72.32
Test 11	98.88	Test 32	96.79	Test 52	99.22
Test 12	85.46	Test 33	98.49	Test 53	92.20
Test 13	93.57	Test 36	97.24	Test 54	97.16
Test 15	82.35	Test 37	99.66	Test 55	74.72
Test 16	97.65	Test 38	97.51	Test 56	98.51
Test 17	97.18	Test 40	99.56	Test 57	99.25
Test 18	98.32	Test 41	90.04	Test 58	97.71
Test 19	99.24	Test 42	85.64	Rata-rata	94.26

Tabel A. 42 Hasil Akurasi Uji Coba 5 (Bagian 1) –linear C 0.1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	83.57	Test 20	99.57	Test 43	93.27
Test 2	98.73	Test 21	99.53	Test 44	97.75
Test 3	93.34	Test 22	98.63	Test 45	95.41
Test 4	96.81	Test 23	99.21	Test 46	99.36
Test 5	96.67	Test 25	84.61	Test 47	95.33
Test 6	93.50	Test 26	99.56	Test 48	78.59
Test7	91.37	Test 27	95.42	Test 49	76.19
Test 9	95.45	Test 29	98.63	Test 50	99.14
Test 10	97.34	Test 31	99.27	Test 51	72.39
Test 11	98.73	Test 32	96.79	Test 52	99.22
Test 12	98.73	Test 33	98.51	Test 53	93.09
Test 13	96.39	Test 36	97.20	Test 54	97.19
Test 15	82.41	Test 37	99.66	Test 55	74.72
Test 16	97.60	Test 38	97.97	Test 56	98.54
Test 17	97.27	Test 40	99.55	Test 57	99.32
Test 18	98.27	Test 41	90.19	Test 58	97.58
Test 19	99.34	Test 42	85.66	Rata-rata	94.45

Tabel A. 43 Hasil Akurasi Uji Coba 5 (Bagian 2) –polinomial(3) C 0.1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	84.02	Test 20	99.54	Test 43	95.59
Test 2	98.32	Test 21	99.47	Test 44	93.76
Test 3	92.07	Test 22	98.95	Test 45	96.07
Test 4	96.62	Test 23	98.48	Test 46	99.11
Test 5	97.13	Test 25	84.53	Test 47	95.28
Test 6	94.66	Test 26	99.48	Test 48	78.05
Test7	91.28	Test 27	95.43	Test 49	76.25
Test 9	95.17	Test 29	97.93	Test 50	99.61
Test 10	97.51	Test 31	99.18	Test 51	72.22
Test 11	98.40	Test 32	98.19	Test 52	99.13
Test 12	81.33	Test 33	98.79	Test 53	94.23
Test 13	94.69	Test 36	97.53	Test 54	97.01
Test 15	84.50	Test 37	99.57	Test 55	74.71
Test 16	96.46	Test 38	97.44	Test 56	98.52
Test 17	97.58	Test 40	99.55	Test 57	99.27
Test 18	98.69	Test 41	90.98	Test 58	97.81
Test 19	99.25	Test 42	85.82	Rata-rata	94.10

Tabel A. 44 Hasil Akurasi Uji Coba Skenario 5 (Bagian 3) –polinomial(4) C 0.1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	84.67	Test 20	99.47	Test 43	95.92
Test 2	98.17	Test 21	99.32	Test 44	95.85
Test 3	91.37	Test 22	98.95	Test 45	96.00
Test 4	96.39	Test 23	98.15	Test 46	98.77
Test 5	97.03	Test 25	84.50	Test 47	95.26
Test 6	94.25	Test 26	99.40	Test 48	76.29
Test7	91.10	Test 27	95.56	Test 49	76.32
Test 9	95.09	Test 29	97.76	Test 50	99.61
Test 10	97.54	Test 31	99.15	Test 51	72.21
Test 11	98.29	Test 32	99.19	Test 52	99.11
Test 12	82.05	Test 33	98.82	Test 53	94.71
Test 13	95.05	Test 36	97.46	Test 54	96.93
Test 15	87.69	Test 37	99.45	Test 55	74.69
Test 16	95.71	Test 38	96.69	Test 56	98.51
Test 17	97.54	Test 40	99.55	Test 57	99.28
Test 18	98.78	Test 41	91.13	Test 58	97.89
Test 19	99.20	Test 42	85.78	Rata-rata	94.15

Tabel A. 45 Hasil Akurasi Uji Coba Skenario 5 (Bagian 4) –rbf(0.25) C 0.1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	88.56	Test 20	98.84	Test 43	96.60
Test 2	98.29	Test 21	99.22	Test 44	99.70
Test 3	41.17	Test 22	99.09	Test 45	91.87
Test 4	96.83	Test 23	96.34	Test 46	99.46
Test 5	92.34	Test 25	84.52	Test 47	92.53
Test 6	71.63	Test 26	99.54	Test 48	80.57
Test7	97.32	Test 27	95.41	Test 49	97.66
Test 9	94.79	Test 29	99.10	Test 50	97.31
Test 10	98.90	Test 31	99.11	Test 51	90.16
Test 11	98.30	Test 32	94.58	Test 52	99.13
Test 12	74.31	Test 33	98.76	Test 53	79.87
Test 13	97.28	Test 36	98.44	Test 54	91.43
Test 15	96.46	Test 37	99.52	Test 55	92.72
Test 16	94.04	Test 38	99.03	Test 56	98.54
Test 17	96.72	Test 40	99.56	Test 57	98.83
Test 18	94.15	Test 41	92.09	Test 58	97.31
Test 19	99.37	Test 42	85.37	Rata-rata	93.45

Tabel A. 46 Hasil Akurasi Uji Coba Skenario 5 (Bagian 5) –rbf(0.5) C 0.1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	88.40	Test 20	99.55	Test 43	96.38
Test 2	98.46	Test 21	99.49	Test 44	99.69
Test 3	89.81	Test 22	98.97	Test 45	95.19
Test 4	96.82	Test 23	97.61	Test 46	99.44
Test 5	96.59	Test 25	84.64	Test 47	95.15
Test 6	78.35	Test 26	99.58	Test 48	81.78
Test7	93.30	Test 27	95.54	Test 49	93.38
Test 9	95.54	Test 29	98.95	Test 50	99.22
Test 10	98.29	Test 31	99.27	Test 51	78.61
Test 11	98.65	Test 32	96.77	Test 52	99.23
Test 12	81.55	Test 33	98.78	Test 53	84.78
Test 13	97.76	Test 36	98.78	Test 54	96.88
Test 15	96.57	Test 37	99.65	Test 55	76.22
Test 16	96.98	Test 38	98.84	Test 56	98.56
Test 17	97.23	Test 40	99.57	Test 57	99.27
Test 18	97.82	Test 41	90.84	Test 58	97.53
Test 19	99.48	Test 42	85.68	Rata-rata	94.71

Tabel A. 47 Hasil Akurasi Uji Coba Skenario 5 (Bagian 6) –linear C 1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	87.31	Test 20	99.54	Test 43	93.79
Test 2	98.70	Test 21	99.54	Test 44	95.85
Test 3	91.71	Test 22	98.61	Test 45	95.93
Test 4	96.82	Test 23	99.23	Test 46	99.35
Test 5	97.89	Test 25	84.62	Test 47	95.39
Test 6	95.93	Test 26	99.56	Test 48	81.09
Test7	91.75	Test 27	95.49	Test 49	76.20
Test 9	95.73	Test 29	98.38	Test 50	99.20
Test 10	97.54	Test 31	99.30	Test 51	72.32
Test 11	98.88	Test 32	96.79	Test 52	99.22
Test 12	85.46	Test 33	98.49	Test 53	92.20
Test 13	93.57	Test 36	97.24	Test 54	97.16
Test 15	82.35	Test 37	99.66	Test 55	74.72
Test 16	97.65	Test 38	97.51	Test 56	98.51
Test 17	97.18	Test 40	99.56	Test 57	99.25
Test 18	98.32	Test 41	90.04	Test 58	97.71
Test 19	99.24	Test 42	85.64	Rata-rata	94.26

Tabel A. 48 Hasil Akurasi Uji Coba Skenario 5 (Bagian 7) –polinomial(3) C 1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	85.27	Test 20	99.48	Test 43	96.13
Test 2	98.61	Test 21	99.47	Test 44	99.23
Test 3	88.18	Test 22	98.93	Test 45	96.32
Test 4	96.53	Test 23	98.37	Test 46	99.07
Test 5	97.57	Test 25	84.57	Test 47	95.29
Test 6	94.98	Test 26	99.48	Test 48	77.71
Test7	91.55	Test 27	95.52	Test 49	76.31
Test 9	95.36	Test 29	97.82	Test 50	99.61
Test 10	97.61	Test 31	99.21	Test 51	72.19
Test 11	98.65	Test 32	98.08	Test 52	99.14
Test 12	82.90	Test 33	98.83	Test 53	94.35
Test 13	95.56	Test 36	97.52	Test 54	96.76
Test 15	85.05	Test 37	99.54	Test 55	74.70
Test 16	96.46	Test 38	96.22	Test 56	98.50
Test 17	97.53	Test 40	99.55	Test 57	99.23
Test 18	98.76	Test 41	90.98	Test 58	97.83
Test 19	99.06	Test 42	85.81	Rata-rata	94.23

Tabel A. 49 Hasil Akurasi Uji Coba Skenario 5 (Bagian 8) –polinomial(4) C 1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	84.11	Test 20	99.44	Test 43	96.56
Test 2	98.51	Test 21	99.31	Test 44	99.14
Test 3	85.10	Test 22	98.95	Test 45	96.23
Test 4	96.39	Test 23	97.92	Test 46	98.75
Test 5	97.21	Test 25	84.86	Test 47	95.24
Test 6	92.31	Test 26	99.43	Test 48	77.80
Test7	92.31	Test 27	95.68	Test 49	76.40
Test 9	95.18	Test 29	97.68	Test 50	99.61
Test 10	97.60	Test 31	99.18	Test 51	72.53
Test 11	98.55	Test 32	99.17	Test 52	99.13
Test 12	83.41	Test 33	98.86	Test 53	94.72
Test 13	96.58	Test 36	97.49	Test 54	96.72
Test 15	88.41	Test 37	99.45	Test 55	74.67
Test 16	95.71	Test 38	95.71	Test 56	98.49
Test 17	97.51	Test 40	99.56	Test 57	99.27
Test 18	98.85	Test 41	91.11	Test 58	97.97
Test 19	99.30	Test 42	85.81	Rata-rata	94.20

Tabel A. 50 Hasil Akurasi Uji Coba Skenario 5 (Bagian 9) –rbf(0.25) C 1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	89.28	Test 20	98.74	Test 43	96.73
Test 2	98.09	Test 21	99.22	Test 44	99.40
Test 3	36.83	Test 22	99.09	Test 45	91.89
Test 4	96.83	Test 23	97.76	Test 46	99.46
Test 5	92.65	Test 25	84.46	Test 47	92.38
Test 6	73.90	Test 26	99.53	Test 48	80.24
Test7	97.39	Test 27	95.38	Test 49	98.00
Test 9	94.77	Test 29	99.10	Test 50	97.22
Test 10	98.93	Test 31	99.12	Test 51	90.14
Test 11	98.22	Test 32	94.72	Test 52	99.14
Test 12	74.31	Test 33	98.75	Test 53	79.81
Test 13	96.68	Test 36	98.45	Test 54	91.12
Test 15	96.49	Test 37	99.53	Test 55	93.18
Test 16	94.06	Test 38	94.86	Test 56	98.48
Test 17	96.67	Test 40	99.56	Test 57	98.79
Test 18	94.11	Test 41	92.12	Test 58	97.32
Test 19	99.37	Test 42	85.34	Rata-rata	93.35

Tabel A. 51 Hasil Akurasi Uji Coba Skenario 5 (Bagian 10) –rbf(0.5) C 1

Nama Citra	Akurasi	Nama Citra	Akurasi	Nama Citra	Akurasi
Test 1	87.53	Test 20	99.50	Test 43	97.14
Test 2	98.56	Test 21	99.49	Test 44	99.47
Test 3	71.15	Test 22	99.01	Test 45	94.64
Test 4	96.83	Test 23	97.75	Test 46	99.43
Test 5	96.46	Test 25	84.58	Test 47	95.17
Test 6	78.93	Test 26	99.58	Test 48	82.36
Test7	93.64	Test 27	95.51	Test 49	95.17
Test 9	95.67	Test 29	98.87	Test 50	99.27
Test 10	98.43	Test 31	99.27	Test 51	79.62
Test 11	98.81	Test 32	96.76	Test 52	99.22
Test 12	80.70	Test 33	98.79	Test 53	88.41
Test 13	97.19	Test 36	98.07	Test 54	96.56
Test 15	96.45	Test 37	99.66	Test 55	96.56
Test 16	96.97	Test 38	98.79	Test 56	98.53
Test 17	97.32	Test 40	99.57	Test 57	99.27
Test 18	97.79	Test 41	90.92	Test 58	97.53
Test 19	99.49	Test 42	85.65	Rata-rata	94.84

DAFTAR PUSTAKA

- [1] A. Y. W. S. M. D. A. N. S. M. Dinda Firly, Impelementasi Support Vector Machine dengan Optimasi Pruning pada Persoalan Klasifikasi Pola, Surabaya: Institut Teknologi Sepuluh Nopember, 2014.
- [2] J. S. N.-N. Z. X. T. a. H.-Y. Tie Liu. [Online]. Available: http://research.microsoft.com/en-us/um/people/jjiansun/SalientObject/salient_object.htm. [Accessed 03 June 2015].
- [3] X. H. a. L. Zhang, "Saliency Detection: A Spectral Residual Approach," 2007.
- [4] "Wikipedia The Free Encyclopedia," 22 May 2015. [Online]. Available: http://en.wikipedia.org/wiki/Image_segmentation. [Accessed 4 June 2015].
- [5] "MathWorks," [Online]. Available: <http://www.mathworks.com/help/images/ref/graythresh.html>. [Accessed 4 June 2015].
- [6] W. W. Xuafei Bai, "Saliency-SVM: An automatic approach for image segmentation," *Neurocomputing*, vol. 136, pp. 243-255, 2014.
- [7] Q.-Y. W. H.-Y. Y. J. B. Xiang-Yang Wang, "Color image segmentation using automatic pixel classification with support vector machine," *Neurocomputing*, vol. 74, pp. 3898-3911, 2011.
- [8] D. E. N. S. S. M. D. P. S. M. Niswah Atiqotun, Temu Kembali Citra Berbasis Isi Pada Citra Kain berdasarkan Fitur Warna, Tekstur, dan Bentuk, Surabaya: Institut Teknologi Sepuluh Nopember, 2014.
- [9] J. Pang, "Matlab Central File Exchange," 07 January 2014. [Online]. Available:

<http://www.mathworks.com/matlabcentral/fileexchange/44956-steerable-filter>. [Accessed 29 Mei 2015].

- [10 "MathWorks Documentation," MathWorks, [Online].
] Available:
<http://www.mathworks.com/help/stats/svmtrain.html>.
[Accessed 2 June 2015].
- [11 "MathWorks Documentation : svmclassify," MathWorks,
] [Online]. Available:
<http://www.mathworks.com/help/stats/svmclassify.html>.
[Accessed 2 June 2015].

BIODATA PENULIS



Aisha Yuliandari, penulis dari buku Tugas Akhir ini lahir di kota Pekanbaru pada tanggal 13 Juli 1993. Penulis telah menempuh pendidikan di SD Cendana Rumbai-Pekanbaru (1999-2005), SMP Cendana Rumbai-Pekanbaru (2005-2008), SMA Cendana Rumbai-Pekanbaru (2008-2011) dan terakhir di Teknik Informatika ITS Surabaya (2011-2015). Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Komputer (HMTK). Diantaranya menjadi staff departemen Hubungan Luar teknik computer ITS 2012-2013. Penulis juga aktif dalam kegiatan kepanitiaan Schematics, diantaranya pernah menjadi Bendahara 2 Schematics pada tahun 2012 dan menjadi staf Revolutionary Expo with Various Arts (REEVA) tahun 2013. Penulis juga aktif pada Tari Saman Teknik Informatika ITS pada tahun 2012-2014. Selama kuliah di teknik informatika ITS, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV) tertarik dengan topik pengolahan citra dan *machine learning*. Penulis dapat dihubungi melalui email, aishayuliandari@gmail.com.